Ph.D. DISSERTATION

Blind Image Deblurring using Kernel-guided Nonlocal Patches and Nonlocal Low-rank Images

커널에 의한 비근접 부분영상과 저차수 영상을 이용한 영상 선명화 기법

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Suil Son

Ph.D. DISSERTATION

Blind Image Deblurring using Kernel-guided Nonlocal Patches and Nonlocal Low-rank Images

커널에 의한 비근접 부분영상과 저차수 영상을 이용한 영상 선명화 기법

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Suil Son

Blind Image Deblurring using Kernel-guided Nonlocal Patches and Nonlocal Low-rank Images

커널에 의한 비근접 부분영상과 저차수 영상을 이용한 영상 선명화 기법

지도교수 유 석 인

이 논문을 공학박사학위논문으로 제출함

2015 년 10 월

서울대학교 대학원

전기·컴퓨터 공학부

손 수 일

손수일의 박사학위논문을 인준함

2015 년 12 월

위 원 장 _____ (인)
부위원장 _____ (인)
위 원 _____ (인)
위 원 _____ (인)
위 원 _____ (인)

Abstract

Blind Image Deblurring using Kernel-guided Nonlocal Patches and Nonlocal Low-rank Images

Suil Son
Department of Electrical Engineering and Computer Science
College of Engineering
The Graduate School
Seoul National University

Blind image deblurring aims to restore a high-quality image from a blurry image. Blind image deblurring has gained considerable attention in recent years because it involves many challenges in problem formulation, regularization, and optimization. In optimization perspective, blind image deblurring is a severely ill-posed inverse problem; therefore, effective regularizations are required in order to obtain a high-quality latent image from a single blurred one. In this paper, we propose nonlocal regularizations to improve blind image deblurring. First, we propose to use the nonlocal patches selected by similarity weighted by the kernel for the next blur-kernel estimation. Using these kernel-guided nonlocal patches, we impose regularization that nonlocal patches would produce the similar values by convolution. Imposing this regularization improves the kernel estimation. Second, we propose to use a nonlocal low-rank image obtained from the composition of nonlocal similar patches. Using this nonlocal low-rank image, we impose regularization that the latent image is similar to this nonlocal low-rank image. A nonlocal low-rank image contains less noise by

its intrinsic property. Imposing this regularization improves the estimation of the latent image with less noise. We evaluated our method quantitatively and qualitatively by comparing several conventional blind deblurring methods. For the quantitative evaluation, we computed the sum of squared error, peak signal-to-noise ratio, and structural similarity index. For blurry images without noise, our method was generally superior to the other methods. Especially, the results of ours were sharper on structures and smoother on flat regions. For blurry and noisy images, our method highly outperformed the conventional methods. Most of other methods could not successfully estimate the blur-kernel, and the image blur was not removed. On the other hand, our method successfully estimate the blur-kernel by overcoming the noise and restored a high-quality of deblurred image with less noise.

Keywords: Blind image deblurring, nonlocal regularization, nonlocal low-rank prior, blur-kernel estimation

Student Number: 2008-22937

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image deblurring is a classical and challenging inverse problem which aims to recover a high-quality image from a blurry-noisy one. It has gained considerable attention in recent years because it involves many challenges in problem formulation, regularization, and optimization. Images can be blurry for a number of reasons. For example, a camera might have been shaken during exposure, and the image can be corrupted by the motion of the camera. This is known as a motion blur. The formation process of such image blurs is usually modeled as

$$B = k \otimes I + n, \tag{1.1}$$

where B is the blurred image, k is the blur kernel, I is the latent image, and n is the additive noise. The operator $\otimes$ denotes the convolution operator. The blur kernel, which is also known as the *point-spread function* (PSF), represents the motion of a camera during exposure.

Image deblurring is the inverse of this process. Nonblind image deblurring is to obtain the latent image I when the blurry image B is observed and the blur kernel k is known:

$$\min_I \|B - k \otimes I\|_2. \tag{1.2}$$

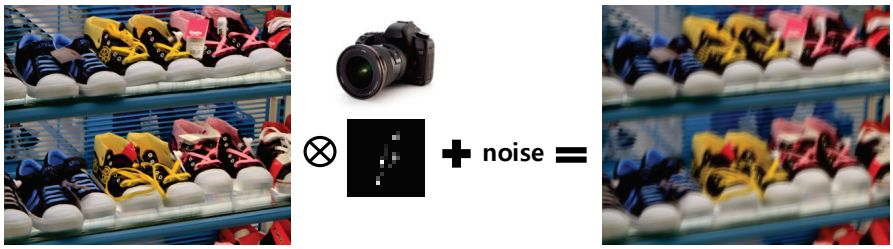


Figure 1.1 The process of image blurs.

This is nonblind in the sense that the blur kernel k is known. If the blur kernel k is also unknown, we have to estimate I and k simultaneously:

$$\min_{I,k} \|B - k \otimes I\|_2. \tag{1.3}$$

This is known as a blind image deblurring in the sense that the blur kernel k is also unknown. The blind image deblurring is a severely ill-posed problem. Given an observed blurry image B, there are infinitely many pairs of k and I that satisfy Eq. (1.1). For example, in Figure 1.2, three pairs of an latent image and a blur kernel are possible solutions. To obtain reasonable solution of the blind image deblurring, the application of prior knowledge of a latent image and a blur kernel is essential.

1.1 Formulation of the Blind Image Deblurring

Blind deblurring can be represented by a probabilistic formulation. In the Bayesian framework, this formulation corresponds to maximizing the posterior distribution, MAP,

$$\arg\max_{I,k} p(I,k|B) = \arg\max_{I,k} p(B|I,k)p(I)p(k), \tag{1.4}$$

where $p(B|I,k)$ models the noise affecting the blurry image; $p(I)$ models the distribution of typical sharp images; and $p(k)$ is the prior knowledge on the

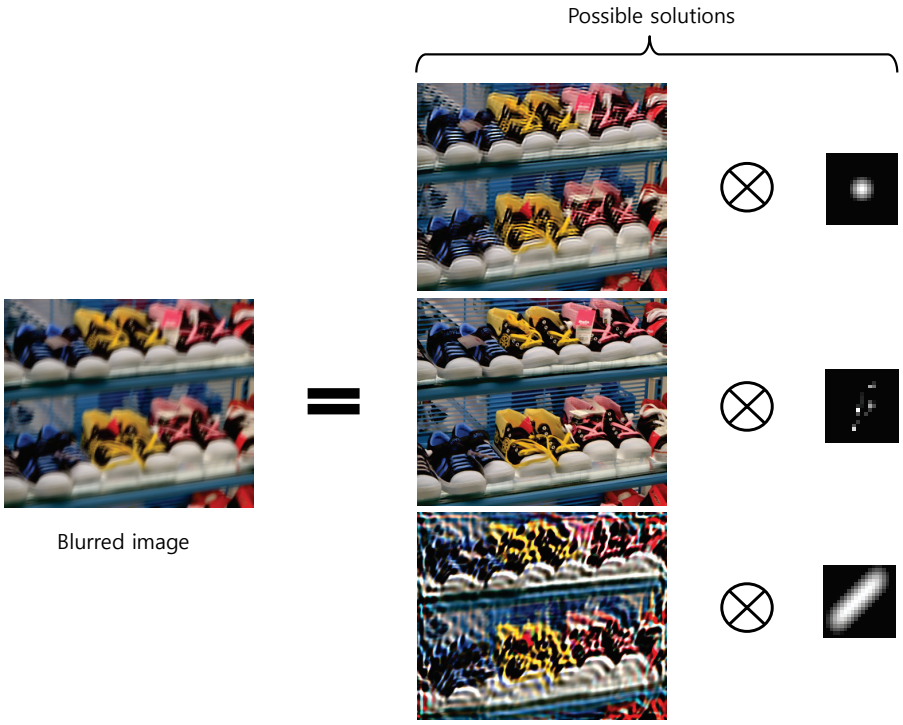


Figure 1.2 The ill-posedness of the image deblurring.

PSF. $p(B|I,k)$ was usually modeled by the Gaussian distribution [31, 46] or an exponential distribution [81]; $p(I)$ was by a heavy-tailed distribution of the image gradients; and $p(k)$ was by a Gaussian distribution [81, 19], a sparsity-including distribution [31, 68] or a uniform distribution [46]. These priors on the latent image I and the blur-kernel k are very important since obtaining the latent image and the kernel is severely ill-posed inverse problem,

In the optimization perspective, if the formation process of a blurry image B is modeled as (1.1), blind deblurring is to solve the following regularized minimization under Gaussian noise assumption:

$$\min_{I,k} \|B - k \otimes I\|_2 + \lambda J(I) + \gamma G(k), \tag{1.5}$$

where the first term is a data fitting term which enforces the similarity between the observed image B and the convolution of the kernel k and the latent image

I, $J(I)$ and $G(k)$ are regularizations that correspond to the priors for I and k, and λ and γ are nonnegative parameters. To solve this optimization, additional constraints on k, such as positivity of its entries and sum to 1, are usually included.

1.2 Approach

Motivated by the recent advances in nonlocal image restoration [87], sparse representation and self-similarity property of images have been successfully applied to the problems of image restoration. In these methods, images are decomposed into small patches, and sparse linear combination of those patches reconstruct the images [27, 9]. Low rank approximation of a matrix can also be applied those patches for enhancing the quality of images. The works [22, 77] provided the connection among the sparse representation, nonlocal self-similarity, and low rank approximation. Motivated by these works, in this paper, we propose to use the nonlocal property of images for regularizations of blind image deblurring.

1.2.1 The Use of Kernel-guided Nonlocal Patches

For the blur kernel estimation, we propose to use nonlocal patches selected by the similarity weighted by the kernel. Ordinary nonlocal similar patches have possibility of resulting in quite different values under convolution operation because the convolutional kernel amplifies or reduces the effect of each element depending on its values. We select nonlocal similar patches based on the similarity weighted by the kernel. If the estimated kernel is correct, those nonlocal patches could lead to the similar value by convolution. Therefore, the weighted nonlocal patches can be used to improve the kernel estimation. In addition, previous works [58, 59] show that the salient edges help kernel estimation. We therefore utilize the salient edges not only to estimate the blur kernel but also to extract candidate nonlocal patches for kernel estimation.

1.2.2 The Use of Nonlocal Low-rank Images

Nonlocal patches can also be used for the estimation of a latent image. We propose to use the nonlocal low-rank image as prior knowledge in order to suppress noise and to obtain high-quality images even for the blurred images with noise. The estimation of the latent image is the final goal of the deblurring; meanwhile, an interim image is used as data for the estimation of the blur kernel. In the estimation of the blur kernel, noise and small structures of the interim images are obstacles of the estimation of the correct kernel. Even when the correct kernel is estimated, the high-quality latent image could not be obtained if the blurred image contains strong noise. If we employ the property of the nonlocal self-similarity in natural images, we can construct a nonlocal low-rank image from a single image by using the low-rank property of the nonlocal similar patches [22, 23]. The regularization of this nonlocal low-rank image guides the estimation of the latent image to be less noisy.

1.3 Overview

The remainder of this paper is organized as follows. We firstly visit the related works in Chapter 2. In Chapter 3, we study the preliminary of optimization which is required for algorithm in this paper. We explain the method to extract salient structures in Chapter 4. In Chapter 5, we show how to use the salient edges and the nonlocal patches for kernel estimation and to use the nonlocal low-rank images for the estimation of the interim and latent images. We illustrate the experimental results of our approach with comparison to other conventional methods in Chapter 6. Finally, we conclude in Chapter 7.

Chapter 2

Related Works

Much research has been devoted to blind image deblurring. Since the blind image deblurring is an ill-posed problem, there exist many solutions to Eq. (1.3). To obtain a high-quality image as a solution, appropriate regularizations are necessary. The representative regularization is a natural image prior [31, 44]. For the same reason of the ill-posedness, a no-blur solution is another difficulty in blind image deblurring. A no-blur solution of delta kernel and the given blurry image is also a solution of Eq. (1.3). This is obtained without appropriate regularizations and careful optimization scheme. To overcome this phenomenon, Krishnan *et al.* [40] and Levin *et al.* [46] have proposed various approaches. Many works utilizing the sparse representation also have been proposed. The core of these approaches is to restore images by the sparse linear combination of the elements of a dictionary which is learned in advance. The sparse coding of those elements enabled the restoration of clear images even when it lacked the information on latent images. In addition, blind image deblurring has suffered from noise. Motion blur usually occurs under long exposure of a camera shutter because of the lack of light. This situation causes much noise, and the noise in blurry images severely spoils the kernel estimation. Therefore, deblurring

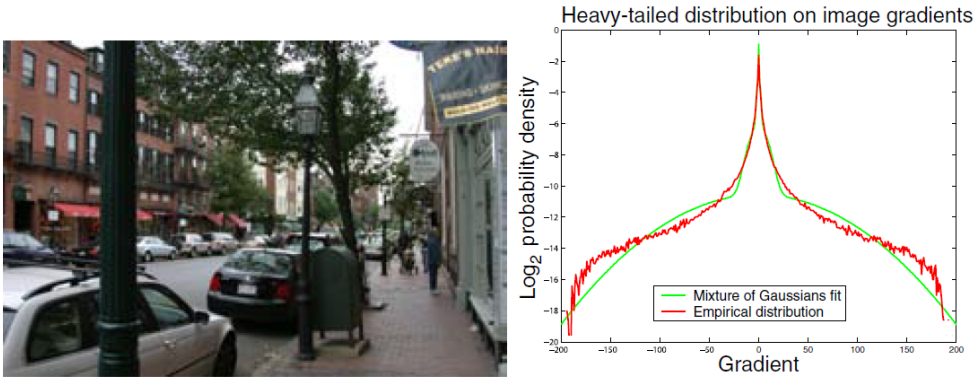


Figure 2.1 Heavy-tailed distribution on image gradients [31].

overcoming noise is a big challenge.

2.1 Natural Image Prior

For the estimation of the high-quality latent image from limited observations as in blind image deblurring, it is essential to have knowledge that explains which images are more likely. Recent research in natural image statistics have shown that, although absolute color of images is distributed over various values, their gradients follow heavy-tailed distributions: the distribution of gradients has most of its mass on small values but gives significantly more probability to large values than a Gaussian distribution. This coincides our intuition that images often contain large sections of constant intensity or gentle intensity gradient and large changes occasionally occur at edges or occlusion boundaries. For example, Fig. 2.1 shows a natural image and a histogram of its gradient magnitudes. The distribution shows that most of the the image gradients are small or zero, but a few gradients have large magnitudes. Many image processing methods based on heavy-tailed distribution gave state-of-the-art results in image denoising [64, 69, 71] and superresolution [75]. In contrast, methods based on Gaussian prior distributions produce overly smooth images.

2.1.1 Scale Mixture of Gaussians

To fit the distribution of natural image gradients, Fergus *et al.* [31] adopted a zero mean mixture of Gaussians. Given the gray-scale blurred patch $\mathbf{P}$, they estimated the blur-kernel $\mathbf{K}$ and the latent patch image $\mathbf{L}_p$ as finding the values with highest probability, guided by a prior on the statistics of $\mathbf{L}$. Since the prior is on the image gradients, they performed the optimization in the gradient domain rather than the intensities, using $\nabla\mathbf{L}_p$ and $\nabla\mathbf{P}$, the gradient of $\mathbf{L}_p$ and $\mathbf{P}$. Because convolution is a linear operation, the patch gradients $\nabla\mathbf{P}$ still equal to the convolution of the latent gradients and the kernel: $\nabla\mathbf{P} = \nabla\mathbf{L}_p \otimes \mathbf{K}$, plus noise. The prior $p(\nabla\mathbf{L}_p)$ on the latent image gradients was modeled as a mixture of C zero-mean Gaussians with variance v_c and weight π_c for the c-th Gaussian. A sparsity prior $p(\mathbf{K})$ was used for the kernel that encourages zero values in the kernel. All entries of the kernel are required to be positive. Specifically, the prior on kernel values is a mixture of D exponential distributions with scale factors λ_d and weights π_d for the d-th component. Given the measured image gradients $\nabla\mathbf{P}$, the posterior distribution over $\mathbf{K}$ and $\nabla\mathbf{L}_p$ is

$$\begin{aligned} p(\mathbf{K}, \nabla\mathbf{L}_p|\nabla\mathbf{P}) &\propto p(\nabla\mathbf{P}|\mathbf{K}, \nabla\mathbf{L}_p)p(\nabla\mathbf{L}_p)p(\mathbf{K}) \\ &= \prod_i \mathcal{N}(\nabla\mathbf{P}(i)|\mathbf{K} \otimes \nabla\mathbf{L}_p(i), \sigma^2) \\ &\quad \prod_i \sum_{c=1}^{C} \pi_c \mathcal{N}(\nabla\mathbf{L}_p(i)|0, v_c) \prod_j \sum_{d=1}^{D} \pi_d \mathcal{E}(\mathbf{K}_j|\lambda_d), \end{aligned} \tag{2.1}$$

where i indexes over image pixels and j indexes over blur kernel elements. $\mathcal{N}$ and $\mathcal{E}$ denote Gaussian and Exponential distributions, respectively. The gradients in $\nabla\mathbf{P}$ were assumed to be independent of each other for tractability, as were the elements in $\nabla\mathbf{L}_p$ and $\mathbf{K}$.

2.1.2 Hyper-Laplacian Distribution

Another approach to modeling the natural image prior is to use a Hyper-Laplacian distribution. Gradient priors are typically enforced between neigh-

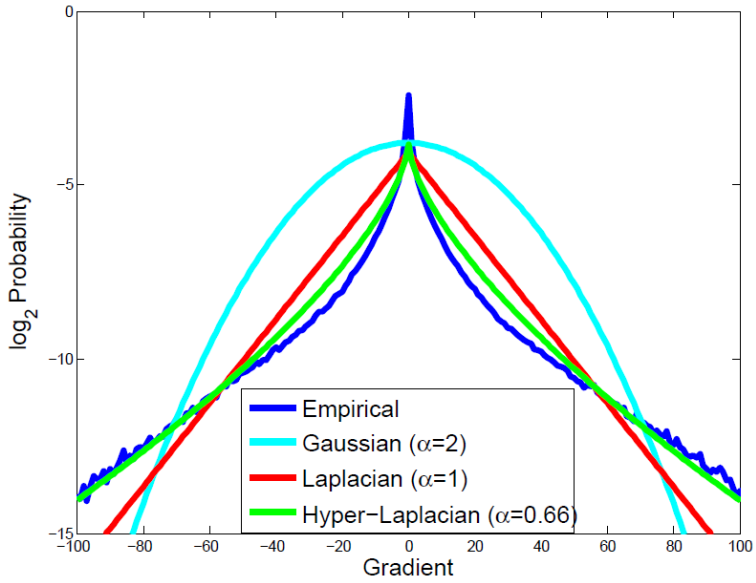


Figure 2.2 Hyper-Laplacian prior [39].

boring pixels in an image. The interactions of such neighboring pixels can be modeled using a Markov Random Field (MRF) in which the value of an individual pixel is conditionally dependent on the pixel values in a local neighborhood. A smoothness prior on the gradients is one possible prior. Since large image gradients are penalized in this prior, neighboring pixels are favored to have values similar to their neighbors. This prior is typically enforced under an assumption of a Gaussian distribution on the image gradients. If we use a Gaussian image prior, the optimal solution can be derived in a closed form. While such a prior does disambiguate the solution, it can result in an overly-smooth solution and introduce ringing artifacts [3]. This stems from the quadratic penalty term, which enforces a Gaussian distribution on gradients. However, the distribution of Gradients in natural images is sparse and not Gaussian. The advantage of a sparse prior for several image processing applications has been demonstrated by numerous recent works [63, 79, 31, 64, 44]. Sparse prior gives more favors to the large magnitudes of gradients than a Gaussian prior, while leaving the majority

of image pixels constant. For example, Fig. 2.2 shows that the Hyper-Laplacian distribution models the natural image statistics. Levin *et al.* [43] addressed this by modifying the gradient penalty to enforce a hyper-Laplacian distribution:

$$L(I) = \lambda \|\nabla I\|^{0.8}. \tag{2.2}$$

The value ∇I indicates the spatial gradients of the image, and λ is a regularization parameter that controls the weight of the smoothness penalty. This sparse prior not only produces sharper edges but also reduces noise and unwanted image artifacts such as ringing. However, the sparse prior induces a non-convex optimization problem which cannot be minimized in a closed form. To optimize this, Levin *et al.* used an iterative re-weighted least squares process (IRLS) [44].

2.2 Avoiding No-blur Solution

The image deconvolution is a severely ill-posed problem, in which there is an infinite set of pairs of a latent image and a blur-kernel which can explain an observed blurry image. The infinite numbers of solutions include some undesirable solutions. For example, the no-blur solution, in which k is the delta (identity) kernel and $I = B$, perfectly satisfies the Eq. (1.1). To avoid this no-blur solution, additional assumptions on I or k must be introduced. Those assumptions can be imposed using MAP approach. However, Levin *et al.* [45] argued that typical MAP approach cannot avoid the no-blur solution as well. According to them, even the use of a sparse derivative prior that favors sharp images cannot prevent it. A direct application of this principle has not yielded the expected results and many algorithms have required additional components, such as marginalization across all possible images [52, 31, 42], spatially-varying terms [36, 68], or solvers that vary their optimization energy over time [68].

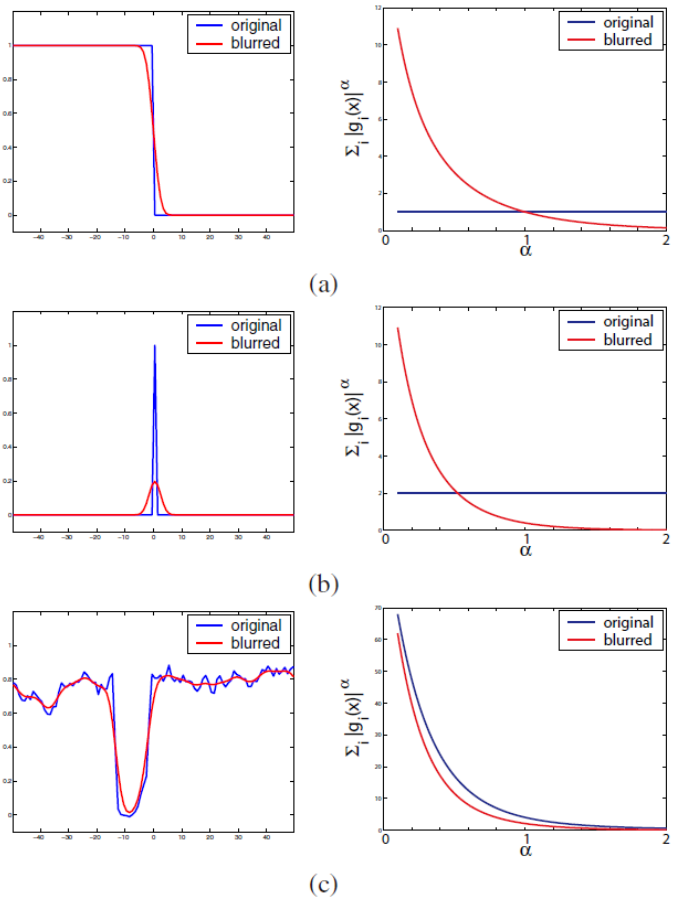


Figure 2.3 The $\text{MAP}_{x,k}$ score evaluated on toy 1D signals [45].

2.2.1 Marginalization over Possible Images

Levin *et al.* [45] analyzed the failure of the $\text{MAP}_{x,k}$ approach:

$$(x, k) = \arg\max_{x,k} \lambda \|k \otimes x - y\|_2^2 + \|\nabla x\|_\alpha^\alpha \tag{2.3}$$

They considered the 1D signals x in Fig. 2.3 which were convolved with a truncated Gaussian kernel k^* of standard deviation 4 pixels. In this figure, $-\log p(x, k|y)$ score was computed using the true kernel and the delta kernel and was compared, while varying the α parameter in the prior. $\text{MAP}_{x,k}$ approach was usually successful for step edges as shown in Fig. 2.3 (a). The edge was

sharper than that of blurred version, and appropriate sparse prior ($\alpha < 1$) favored the correct sharp explanation, while the Gaussian prior favored the blurry one. On the contrary, for the signal of a narrow peak in Fig. 2.3 (b), blurring reduces the peak height, and the Laplacian prior $\alpha = 1$ favored the blurry x. Examination of Fig. 2.3 (b-right) shows that the blurred explanation is winning for smaller α values as well. Only for very low alpha values, the sharp explanation is favored although the sparse models describing natural images are usually in the range $\alpha \in [0.5, 0.8]$ [70]. The last signal considered in Fig. 2.3 (c) is a row cropped from a natural image. The narrow peaks of this figure correspond to the texture and noise of natural images. This row illustrates that natural images contain a lot of medium contrast texture and noise. These narrow peaks dominate the statistics more than step edges. Therefore, blurring a natural image reduces the overall contrast and, as in Fig. 2.3 (b), even though sparse priors favor the blurry x explanation. Levin *et al.* [45] analyzed these results. Blur has two opposite effects on the image likelihood: 1) it makes the signal derivatives less sparse, and this reduces the likelihood. 2) it reduces the derivative variance and this increases its likelihood. For very specific images, like ideal step edges, the first effect dominates and blur reduces the likelihood. However, for most natural images the second effect is stronger and blur increases the likelihood.

To overcome the limitation of $\text{MAP}_{x,k}$ approach, Levin *et al.* [45] suggested an efficient marginal likelihood approximation which solves the following MAP_k problem

$$\arg\max_k p(k|y) = \arg\max_k \int p(y|x,k)p(x)p(k)dx. \tag{2.4}$$

Then they estimate x by solving a convex problem where k is given from the previous step. Estimation theory and statistical signal processing have pointed out the limitation of MAP estimation in the case of few measurements [72, 7]. In the $\text{MAP}_{x,k}$ problem, enough measurements cannot be collected because the number of unknowns grows with the image size. In contrast, if enough mea-

surements are given, MAP estimators do approach the true solution according to estimation theory [72]. The two unknowns x and k shows a strong asymmetry in their dimensionality. To obtain good MAP estimators, this asymmetric property of blind deconvolution can be utilized. While the dimensionality of x increases with the image size, the support of the kernel is fixed and small relative to the image size. The image y does provide a large number of measurements for estimating k. Thus, a MAP_k estimation of k alone (marginalizing over x) can recover the true kernel with an increasing accuracy.

2.2.2 Normalization of l_1 by l_2

Krishnan *et al.* [40] also visited the failure of $\text{MAP}_{I,k}$ approach. They firstly analyzed the following l_1 regularization problem:

$$\min_{I,k} \|\nabla B - k \otimes \nabla I\|_2 + \lambda \|\nabla I\|_1 + \psi \|k\|_1. \tag{2.5}$$

The l_1 norm is widely used to impose sparsity of signal. Since the l_1 is scale variant, the norm can also be minimized by simply reducing the signal not by imposing the sparsity. This is easily observed in the denoising problems. In an image setting, the l_1 norm is typically used to penalize the high frequency components. As image noise presents itself in the high frequency bands, boosting their l_1 norm, minimizing the norm is a way of denoising the image. However, in the case of image blur, the opposite situation holds since blur attenuates the high frequency bands so reducing their l_1 norm. Consequently, in blind deconvolution, minimizing the l_1 norm on the high frequencies of the image will result in a blurry image and a delta function kernel. This behavior is illustrated in Figure 2.4.

To overcome this, Krishnan *et al.* [40] proposed a normalized sparsity prior, i.e., l_1/l_2 regularizer. In this formulation, the deblurring is to solve the following:

$$\min_{I,k} \|\nabla B - k \otimes \nabla I\|_2 + \lambda \frac{\|\nabla I\|_1}{\|\nabla I\|_2} + \psi \|k\|_1. \tag{2.6}$$

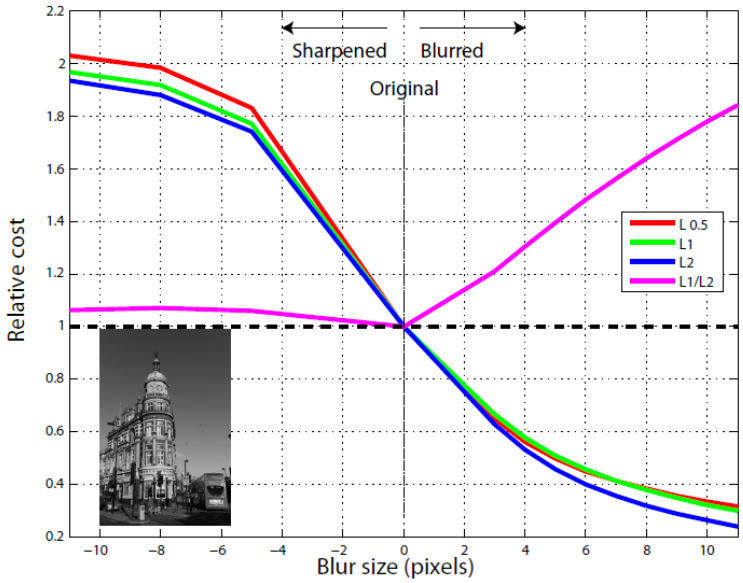


Figure 2.4 Comparison of cost by various regularizers [40].

The l_1/l_2 function is a normalized version of l_1, making it scale invariant. If applied to the high frequency components of an image, it is equivalent to the l_1 norm of the edges rescaled by their total energy. Although blur decreases both of the l_1 and l_2 norms, crucially the latter is reduced more. The ratio of the two, therefore, will be increased by blur. This is also illustrated in Figure 2.4.

To understand this, consider the visualization of the l_1/l_2 function for a two dimensional signal in Figure 2.5. The minima lie along the axes with the cost increasing smoothly in between. The high frequency components of natural scenes are typically sparse in that the magnitudes are mostly either zero or very small, but occasionally large. If these components are represented as a single high dimensional vector, it would be close to the axes in many dimensions and have a low l_1/l_2 value. Increasing the l_1/l_2 value as shown in Figure 2.5 smears out the large magnitude elements and reduces the number of zero elements.

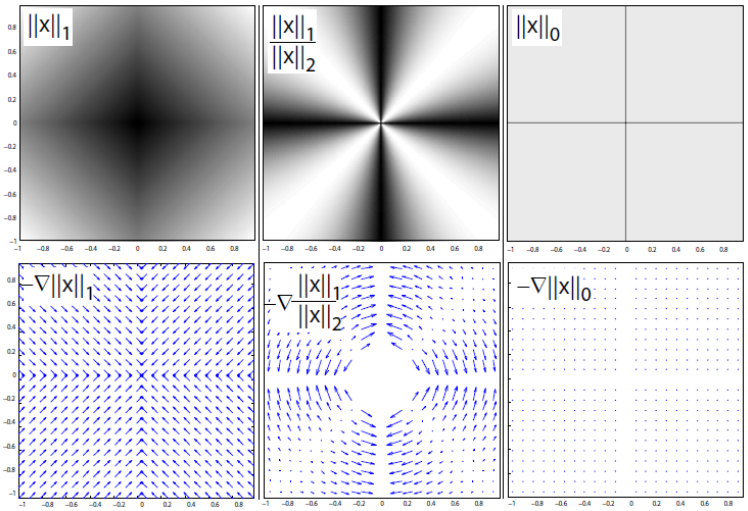


Figure 2.5 Visualization of various regularizers [40].

2.2.3 Alternating I and k Approach

The work by Levin *et al.* [45] has shown that the joint optimization of both image and blur kernel can give the no-blur solution as its global minimum. This means that a wide selection of prior work in blind image deconvolution either is a local minimizer and, hence, requires a good guess about initial solution, or it cannot depart too much from the no-blur solution. Nonetheless, several algorithms based on the joint optimization of blur and sharp image have shown good convergence behavior even when initialized with the no-blur solution [18, 68, 81, 19].

Perrone *et al.* [61] analyzed this paradoxical conundrum in the following formulation:

$$\begin{aligned} &\min_{u,k} \|k \otimes u - f\|_2^2 + \lambda J(u) \\ &\text{subject to } k \succeq 0, \|k\|_1 = 1, \end{aligned} \tag{2.7}$$

where f is an observed blurry image, u is a latent clear image, k is a blur kernel, $J(u) = \|u\|_{BV} \doteq \int \|\nabla u(\mathbf{x})\|_2 d\mathbf{x}$ or $J(u) = \|u_x\|_1 + \|u_u\|_1$, with $\nabla u \doteq [u_x u_y]^T$

the gradient of u and $\mathbf{x} \doteq [xy]^T$.

Levin *et al.* [45] have shown that Eq. 2.7 favors the no-blur solution (f, δ), when $J(u) = \int |u_x(\mathbf{x})|^\alpha + |u_x(\mathbf{u})|^\alpha d\mathbf{x}$, for any $\alpha > 0$ and either the true blur k_0 has a large support of $\|k_0\|_2^2 \ll 1$. A solution to problem Eq. 2.7 can be found by an iterative algorithm that alternates between the estimation of the sharp image given the kernel and the estimation of the kernel given the sharp image. This approach, called *alternating minimization*(AM), solves an unconstrained convex problem in u

$$u^{t+1} \leftarrow \min_u \|k^t \otimes u - f\|_2^2 + \lambda J(u) \tag{2.8}$$

and a constrained convex problem in k

$$\begin{aligned} k^{t+1} \leftarrow & \min_k \|k \otimes u^y - f\|_2^2 \\ & \text{subject to } k \succeq 0, \|k\|_1 = 1. \end{aligned} \tag{2.9}$$

Without a careful initialization, this algorithm could get stuck on the no-blur solution.

To minimize problem (2.7), Chan and Wong [18] employed a gradient descent algorithm for each step and enforced the constraints on the blur kernel in a subsequent step, which is a variant of the AM algorithm. Perrone *et al.* [61] showed that this subsequent normalization step on the blur kernel k enables the algorithm to avoid the no-blur solution. The gradient descent results in the following update for the sharp image u at the t-th iteration

$$u^{t+1} \leftarrow u^t - \epsilon\Big(k_{\_}^t \otimes (k^t \otimes u^t - f) - \lambda\nabla \cdot \frac{\nabla u^t}{\|\nabla u^t\|_2}\Big) \tag{2.10}$$

for some step $\epsilon > 0$ and where $k_{\_}(\mathbf{x}) = k(-\mathbf{x})$. The above iteration is repeated until the difference between the updated and the previous estimate of u are below a certain threshold. The iteration on the blur kernel k is given by

$$k^t \leftarrow k^t - \epsilon(u_{\_}^t \otimes (k^t \otimes u^t - f)), \tag{2.11}$$

where the above iteration is repeated until a convergence criterion is satisfied. The last updated k^t is used to set $k^{t+1/3} \leftarrow k^t$, is satisfied. A sequential projection was applied to impose the the constraints on the blur k:

$$\begin{aligned} k^{t+2/3} &\leftarrow \max\{k^{t+1/3}, 0\}, \\ k^{t+1} &\leftarrow \frac{k^{t+2/3}}{\|k^{t+2/3}\|_1}. \end{aligned} \tag{2.12}$$

This iterative algorithm was called the *projected alternating minimization* (PAM) [61]. The choice of imposing the constraints sequentially rather than during the gradient descent on k seems a rather innocent and acceptable approximation of the correct procedure. According to Perrone *et al.* [61], the PAM does not minimize the original problem 2.7, and it converges to the desired solution. In contrast, the exact AM algorithm cannot leave the no-blur solution (f, δ).

2.3 Sparse Representation

Sparse representation has been successfully used in various image restoration applications [26, 16, 15, 28, 25, 27, 30, 6, 84]. The success of sparse representation owes to the development of the l_1-norm optimization techniques and the fact that natural images are intrinsically sparse in some domains. In wavelet-based image denoising [25], researchers have found that the sparsity of wavelet coefficients can serve as a good prior. This reveals the fact that many types of signals such as natural images can be sparsely represented using a dictionary of atoms, such as discrete cosine transform (DCT) or wavelet bases. When the dictionary is denoted by Φ, we have $x = \Phi\alpha$ and most of the coefficients in α are close to zero, where x is an original image.

Using this sparse representation, Dong *et al.* [24] modeled the image blur process as

$$y = k \otimes \Phi\alpha + n. \tag{2.13}$$

With the sparsity prior, the representation of x over Φ can be estimated from

its observation y by solving the following l_0-minimization problem:

$$\hat{\alpha} = \arg\min_{\alpha} \|y - k \otimes \Phi\alpha\|_2^2 + \lambda\|\alpha\|_0, \tag{2.14}$$

where the l_0-norm counts the number of nonzero coefficients in vector α. Once $\hat{\alpha}$ is obtained, x can then be estimated as $\hat{x} = \Phi\hat{\alpha}$. The l_0-minimization is an NP-hard combinatorial search problem, and is usually solved by greedy algorithms [80, 76]. The l_0-minimization was approximated by a convex l_1-minimization:

$$\hat{\alpha} = \arg\min_{\alpha} \|y - k \otimes \Phi\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{2.15}$$

In addition, recent studies showed that iteratively reweighting the l_1-norm sparsity regularization term can lead to better image restoration (IR) results [17].

In the sparse representation modeling, the choice of dictionary Φ is critial; therefore, much effort has been made in learning a redundant dictionary from a set of example image patches [28, 49, 48, 35, 1, 51, 66, 50, 54, 65, 8]. Given a set of training image patches $S = [s_1, ..., s_N] \in \mathbb{R}^{n\times N}$, the dictionary learning is to jointly optimize the dictionary Φ and the representation coefficient matrix $\Lambda = [\alpha_1, ..., \alpha_N]$ such that $s_i \approx \Phi\alpha_i$ and $\|\alpha_i\|_p \leq T$, where $p = 0$ or 1. This can be formulated by the following minimization problem:

$$(\hat{\Phi}, \hat{\Lambda}) = \arg\min_{\Phi,\Lambda} \|S - \Phi\Lambda\|_F^2 \quad s.t. \quad \|\alpha_i\|_p \leq T, \forall i \tag{2.16}$$

where $\|\cdot\|_F$ is the Frobenius norm. The above minimization problem is non-convex even when $p = 1$. To make it tractable, approximation approaches such as MOD [29] and K-SVD [1] have been proposed to alternatively optimizing Φ and Λ, leading to many state-of-the-art results in image processing [49, 48, 65]. In addition, Cai *et al.* [14] assumed that the latent images and kernels can be sparsely represented by an over-complete dictionary and introduced a framelet and curvelet system to obtain the sparse representation for images and kernels.

2.4 Using Sharp Edges

Due to the loss of information during blurring, the estimation of a blur-kernel, i.e., PSF, from a single blurred image is an inherently ill-posed problem. The observed blurred image provides only a partial constraint on the solution because there are many combinations of PSFs and sharp images that can be convolved to match the observed blurred image. Prior knowledge about the latent image or kernel can resolve the ambiguity of the potential solutions. Early works in this are significantly constrained the form of the kernel [41], while more recently, researchers have put constraints on the underlying sharp image [31]. Once the sharp image is estimated, the blur kernel can also be estimated.

For the successful estimation of the kernel, the edge information on the latent image is important. Therefore, an explicit edge prediction step was employed in various methods to enhance the kernel estimation. Joshi *et al.* [36] proposed to estimate sharp edges that created the blurred observations under the assumption that detected edges were step edges before blurring. In the approach, the location of image features such as edges are detectable even if the feature strength is weakened. They computed sharp edges by first locating step edges and then propagating the local intensity extrema toward the edge. Cho *et al.* [21] detected sharp edges from blurred images directly, and then they employed the Radon transform to estimate the blur kernel. Cho and Lee [19] used bilateral filtering and shock filtering to predict sharp edges iteratively, and then they selected the salient edges for kernel estimation. Xu and Jia [81] adaptively selected useful edges for kernel estimation by using an effective mask computation algorithm. The kernel refinement was achieved by iterative support detection (ISD) method [78]. Hu and Tang [34] learned good regions for kernel estimation and employed a method [19] to estimate kernels. To achieve better selection of edges, Xu *et al.* [83] and Pan and Su [60] used l_0-regularized kernel estimation. Due to the properties of l_0-norm, these methods were able

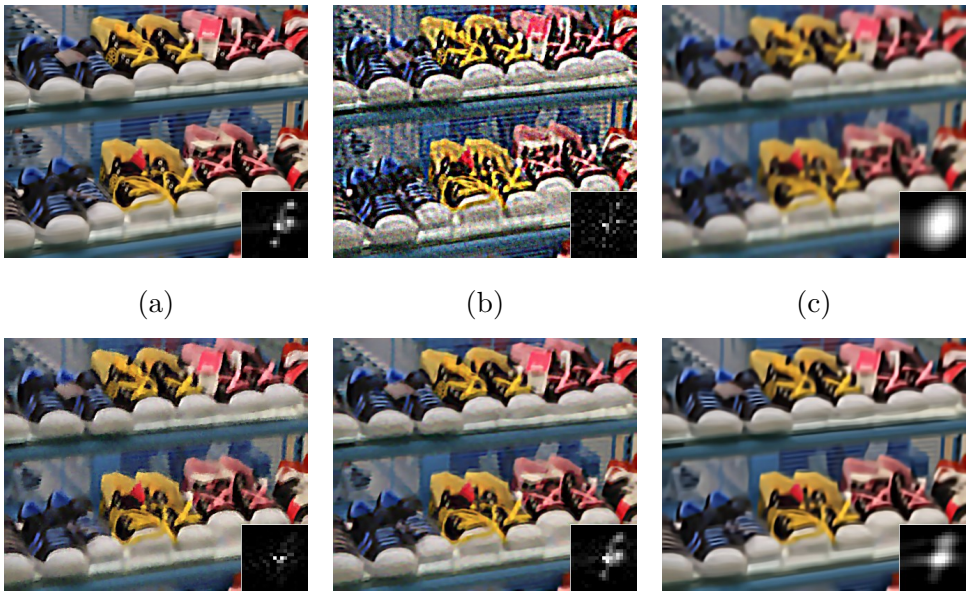
(a) (b) (c)

Figure 2.6 Example of images with blur or noise. (a) Ground truth. (b) Blurry image without noise. (b) Blurry and noisy image.

to select large scale edges for kernel estimation. Sun *et al.* [73] employed patch priors to restore useful edges for kernel estimation. These schemes have been extensively validated in blind image deblurring.

2.5 Handling Noise

Many single image blind deconvolution methods have been proposed [19, 21, 31, 32, 36, 38, 45, 68, 81]. Although they generally work well when the input image is noise-free, their performance degrades rapidly when the noise level increases. The blur kernel estimation was often too fragile to reliably estimate the blur kernel when the image is contaminated with noise as shown in Figure 2.7 (c). Even when an accurate blur kernel is estimated, the estimation of latent image was difficult. The noise was amplified, and ringing artifacts were generated from the non-blind deconvolution [20, 37, 85, 86]. Denoising as a preprocessing step can be considered for the blurry and noisy image. However, denoising damages the useful information in the image. Edges are the main source of information that derives deblurring algorithms either implicitly or explicitly since no information about the blur kernel can be observed in uniform regions of the blurry image. Even small degradation introduced by denoising techniques can have a strong impact on deblurring results as shown in Figure 2.7 (c) - (f).



(a) (b) (c)

(d) (e) (f)

Figure 2.7 The side effect of employing different denoising methods as preprocessing step in image deblurring. Levin *et al.*'s method was applied [46]. (a) Result on blurry image without noise. (b) Result on blurry and noisy image: no preprocessing. (c) Result on blurry image denoised by Gaussian filtering. (d) Result on blurry image denoised by bilateral filtering. (e) Result on blurry image denoised by nonlocal means filtering. (f) Result on blurry image denoised by BM3D

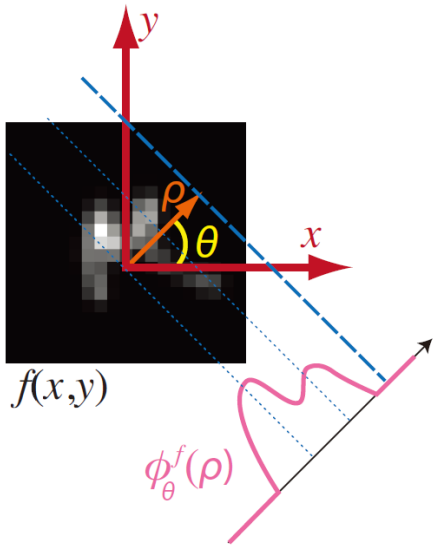


Figure 2.8 The Radon transform $\phi_\theta^f(\rho)$ of a signal f [88].

To handle noisy inputs in single image deblurring, Tai and Lin [74] applied an existing denoising package as preprocessing, and then estimated the blur kernel and the latent image from the denoised result. However, denoising was likely to damage the detailed blur information, thus leading to a biased kernel estimation. Zhong *et al.* [88] showed this. Convolving with a Gaussian G_g decreases the noise level. However, the kernel estimation then becomes:

$$\begin{aligned} k_g &= \arg\min_{k_g} \|B \otimes G_g - I \otimes k_g\|_2^2 \\ &= \arg\min_{k_g} \|(I \otimes k + n) \otimes G_g - I \otimes k_g\|_2^2 \\ &\approx \arg\min_{k_g} \|I \otimes (k \otimes G_g - k_g)\|_2^2 \\ &= k \otimes G_g, \end{aligned} \tag{2.17}$$

where k is the blur kernel for the original input image and k_g is the optimal solution after Gaussian denoising. This shows that the estimated kernel k_g is a blurred version of the actual kernel k. Further, since G_g is a low-pass filter, the high frequencies of k are lost and recovering them from k_g would be very

difficult. This result comes from the initial noise reduction and is independent of the kernel estimation method.

To address this limitation, Zhong *et al.* [88] proposed a method using directional filters and Radon transform. The Radon transform of a signal $f(x, y)$ is the collection of integrals of $f(x, y)$ along straight lines, where the straight lines can be conveniently parameterized using the offset ρ and the orientation θ as shown in Figure 2.8, and it is formulated as

$$\phi_\theta^f(\rho) = \int\int f(x, y)\delta(\rho - x\cos(\theta) - y\sin(\theta))dxdy, \tag{2.18}$$

the function $\phi_\theta^f(\rho)$, for fixed θ, can be viewed as a projection of the signal f along the direction orthogonal to orientation θ. With enough projections of f along different orientations θ, the original signal f can be recovered. This is known as the *inverse Radon transform* and is computationally inexpensive. According to Cho *et al.* [21], the blurred version of horizontal edges B_L produces a vertical Radon projection of the blur-kernel. Mathematically,

$$B_L(x, y) = \int\int k(u, v)\delta(x - u)dudv. \tag{2.19}$$

Zhong *et al.* [88] went further to handle noise. Applying a directional low-pass filter to the input image greatly reduces the noise level, while preserving the blur information in the orthogonal direction to the filter. Based on this observation, they applied a series of directional filters at different orientations to the input image, and estimated an accurate Radon transform of the blur kernel from each filtered image. Then, they reconstructed the blur kernel using inverse Radon transform.

Chapter 3

Preliminary: Optimization

The understanding of the optimization method is essential to blind image deblurring because the blind image deblurring is performed by solving the optimization problems such as Eq. (1.5). In this chapter, we present some optimization schemes of solving the proposed method. Firstly, iterative reweighted least squares (IRLS) [11, 12] is presented. In blind image deblurring, p-norm with $0 < p < 1$ is common. IRLS is effective for solving these p-norm problems. The conjugate gradient method [55] is also presented. The gradient descent method is a fundamental and powerful method of optimization. However, it is slow for data of high dimension and large scale. To overcome these limitations, conjugate gradient method is applied. In our approach, we also utilize the low-rank approach [13]. Among various methods for obtaining the low-rank version of matrices, singluar value thresholding [13] is a simple and effective method. Using these optimization methods, blind image deblurring is numerically achieved.

3.1 Iterative Reweighted Least Squares (IRLS)

Methods of approximating one function by another or of approximating measured data by the output of a mathematical or computer model are extraordinarily useful and ubiquitous. In the formulation of image deconvolution, p-norm with $0 < p < 1$ is prevalent. Many cases of them can be solved by a very powerful algorithm called "Iterative Reweighted Least Squares" (IRLS) [11, 12]. Because minimizing the weighted squared error in an approximation can often be done analytically (or with a finite number of numerical calculations), it is the base of many iterative approaches.

The discussion to IRLS starts with the problem of solving the linear equations:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ a_{21} & a_{22} & a_{23} & & a_{2N} \\ a_{31} & a_{32} & a_{33} & & a_{3N} \\ \vdots & & & \ddots & \vdots \\ a_{M1} & a_{M2} & a_{M3} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_M \end{bmatrix} \tag{3.1}$$

or, in matrix notation

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.2}$$

where $\mathbf{A}$ is an M by N real matrix, $\mathbf{b}$ is an M by 1 vector, and $\mathbf{x}$ is an N by 1 vector, which is what we want to find. Only if $\mathbf{A}$ is non-singular (square and full rank), there is a unique, exact solution. Otherwise, an approximate solution is sought according to some criterion of approximation. If $\mathbf{b}$ does not lie in the column space of $\mathbf{A}$, there is no exact solution to Eq. (3.2); therefore, an approximated solution should be posed. It is obtained by minimizing the norm of an equation error vector defined by

$$\mathbf{e} = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{3.3}$$

The square of the norm of this error vector is also known as a sum of squared errors.

3.1.1 Least Squared Error Approximation

In Eq. (3.2), an $\mathbf{x}$ that minimizes some norm or other measure of $\mathbf{e}$ is considered as a solution. In the case where that problem does not have a unique solution, further conditions such as also minimizing the norm of $\mathbf{x}$ are imposed. The l_2-norm of $\mathbf{e}$ (also known as root-mean-squared error or Euclidean norm) is $\sqrt{\mathbf{e}^T\mathbf{e}}$. The squared error is defined as

$$\|\mathbf{e}\|_2^2 = \sum_i e_i^2 = \mathbf{e}^T\mathbf{e}. \tag{3.4}$$

When this error is minimized, $\mathbf{x}$ is the exact or approximation solution of Eq. (3.2). According to the condition of $\mathbf{A}$, the solution $\mathbf{x}$ is obtained in different manner:

- If $\mathbf{A}$ has $M = N$, (square and nonsingular), then the exact solution is

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{3.5}$$

- If $\mathbf{A}$ has $M > N$, (over specified), then the approximate solution with the least squared equation error is

$$\hat{\mathbf{x}} = [\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{b} \tag{3.6}$$

- If $\mathbf{A}$ has $M < N$, (under specified), then the approximate solution with the least norm is

$$\mathbf{x} = \mathbf{A}^T[\mathbf{A}\mathbf{A}^T]^{-1}\mathbf{b} \tag{3.7}$$

These formulas assume that $\mathbf{A}$ has full row or column rank but, if not, generalized solutions exist using the Moore-Penrose pseudoinverse [2, 4, 10]. The case in Eq. (3.6) is the least squared error approximation of $\mathbf{x}$.

3.1.2 Weighted Least Squared Error Approximation

Least squared error approximation minimizes the l_2 norm of $\mathbf{e}$, and the least norm approximation minimizes the l_2 norm of $\mathbf{x}$. In addition to these cases,

we can consider a more general problem optimizing a weighted norm [4, 5] to emphasize or de-emphasize certain components or range of equations. The linear equation is represented by

$$\mathbf{WAx} = \mathbf{Wb}, \tag{3.8}$$

and the weighted least squared error is

$$\|\mathbf{We}\|_2^2 = \sum_i w_i^2 e_i^2 = \mathbf{e}^T\mathbf{W}^T\mathbf{We}, \tag{3.9}$$

where $\mathbf{W}$ is a diagonal matrix with the weights, w_i, along its diagonal. The approximate solutions [10] of the weighted linear equation are

- If $\mathbf{A}$ has $M > N$, (over specified), then the approximate solution with the least squared equation error is

$$\hat{\mathbf{x}} = [\mathbf{A}^T\mathbf{W}^T\mathbf{WA}]^{-1}\mathbf{A}^T\mathbf{W}^T\mathbf{Wb} \tag{3.10}$$

- If $\mathbf{A}$ has $M < N$, (under specified), then the approximate solution with the least norm is

$$\mathbf{x} = [\mathbf{W}^T\mathbf{W}]^{-1}\mathbf{A}^T\big[\mathbf{A}[\mathbf{W}^T\mathbf{W}]^{-1}\mathbf{A}^T\big]^{-1}\mathbf{b} \tag{3.11}$$

The case in Eq. (3.10) is the weighted least squared error approximation of $\mathbf{x}$.

3.1.3 The l_p Norm Approximation of Overdetermined System

As previously stated, the approximate solutions in Eqs. (3.6) and (3.7) are about l_2 norm. If the objective is about l_p norm such as minimizing $\|\mathbf{e}\|_p^p$, a different approach is necessary. The solutions about the weighted norm in Eqs. (3.10) and (3.11) give a bridge to the l_p norm solutions. Using the analytical solutions of the weighted norm, the IRLS allows an iterative algorithm which applies an iterative reweighting to converge to the optimal l_p approximation [5, 56].

If one poses the l_p approximation problem in solving an overdetermined set of equations, it comes from defining the equation error norm as

$$\|\mathbf{e}\|_p = \Big(\sum_i |e_i|^p \Big)^{1/p} \tag{3.12}$$

and finding $\mathbf{x}$ to minimizing this p-norm of the equation error. This is equivalent to solving a least weighted squared error problem for the appropriate weights [12]:

$$\|\mathbf{e}\|_p = \Big(\sum_i w_i^2 \cdot |e_i|^2 \Big)^{1/2}. \tag{3.13}$$

If one takes Eq. (3.12) and factor the term being summed in the following manner, comparison to Eq. (3.13) suggests the iterative reweighted least square algorithm:

$$\|\mathbf{e}\|_p = \Big(\sum_i |e_i|^{p-2} \cdot |e_i|^2 \Big)^{1/2}. \tag{3.14}$$

To find the minimum l_p approximate solution, the iterative reweighted least squared error algorithm can be used. The IRLS starts with identity weighting, $\mathbf{W} = \mathbf{I}$, solves for an initial $\mathbf{x}$ with Eq. (3.10), calculates a new error from Eq. (3.3), which is then used to set a new weighting matrix $\mathbf{W}$ with diagonal elements of

$$w_i = e_i^{(p-2)/2}, \tag{3.15}$$

which is used in the next iteration of Eq. (3.10) [12]. This is repeated until convergence.

3.1.4 The l_p Norm Approximation of Underdetermined System

IRLS algo gives a method to solving the l_p norm approximation of underdetermined system. For the underdetermined set of equations, l_p approximation comes from defining the solution norm as

$$\|\mathbf{x}\|_p = \Big(\sum_i |x_i|^p \Big)^{1/p} \tag{3.16}$$

and finding $\mathbf{x}$ to minimizing this p-norm while satisfying $\mathbf{Ax} = \mathbf{b}$. It also has been shown that this is equivalent to solving a least weighted norm problem for specific weights:

$$\|\mathbf{x}\|_p = \Big(\sum_i w_i^2 \cdot |x_i|^2 \Big)^{1/2} \tag{3.17}$$

To find this l_p-norm solution, IRLS can be also used. The IRLS starts with identity weighting, $\mathbf{W} = \mathbf{I}$, solves for an initial $\mathbf{x}$ with Eq. (3.11), which is then used to set a new weighting matrix $\mathbf{W}$ with diagonal elements of

$$w_i = x_i^{(2-p)/2}. \tag{3.18}$$

This is used in the next iteration of Eq. (3.11). The iterative procedure is repeated until convergence.

3.2 Optimization using Conjugacy

Conjugate gradient methods are the most useful techniques for solving large linear systems of equations. They can also be adapted to solve nonlinear optimization problems. The linear conjugate gradient method was proposed by Hestenes and Stiefel in the 1950s as an iterative method for solving linear systems with positive definite coefficient matrices [55]. It is an alternative to Gaussian elimination. The performance of the linear conjugate gradient method is determined by the distribution of the eigenvalues of the coefficient matrix. By transforming, or preconditioning, the linear system, we can make this distribution more favorable and improve the convergence of the method significantly. Therefore, preconditioning plays a crucial role in the design of practical conjugate gradient strategies.

The conjugate gradient method is an iterative method for solving a linear system of equations

$$Ax = b, \tag{3.19}$$

where A is an $n \times n$ symmetric positive definite matrix. This problem (3.19) can be stated equivalently as the following minimization problem:

$$\min \phi(x) \equiv \frac{1}{2} x^T A x - b^T x, \tag{3.20}$$

that is, both (3.19) and (3.20) have the same unique solution. For this equivalence, the conjugate gradient method is interpreted either as an algorithm for solving linear systems or as a technique for minimizing convex quadratic functions. The residual of the linear system is denoted as the gradient of ϕ:

$$\nabla \phi(x) = Ax - b \equiv r(x). \tag{3.21}$$

Under this notation, in particular at $x = x_k$, the residual is

$$r_k = Ax_k - b. \tag{3.22}$$

This notation is useful in the following explanation.

3.2.1 The Conjugate Direction Method

To understand the conjugate gradient method, it is convenient to firstly visit the conjugate direction method. A set of nonzero vectors $\{p_0, p_1, ..., p_l\}$ is said to be *conjugate* with respect to the symmetric positive definite matrix A if

$$p_i^T A p_j = 0, \quad \forall i \neq j. \tag{3.23}$$

It is easy to show that any set of vectors satisfying this property is also linearly independent. One of the remarkable properties of the conjugate gradient method is its ability to generate, in a very economical fashion, this set of vectors with property of *conjugagy*.

The importance of conjugacy lies in the fact that we can minimize $\phi(\cdot)$ in n steps by successively minimizing it along the individual directions in a conjugate set. To verify this claim, we consider the following *conjugate direction*

method. Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, ..., p_{n-1}\}$, let us generate the sequence $\{x_k\}$ by setting

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3.24}$$

where α_k is the one-dimensional minimizer of the quadratic function $\phi(\cdot)$ along $x_k + \alpha p_k$, given explicitly by

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}. \tag{3.25}$$

Then, we have the following result [55]:

Theorem 3.2.1. *For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm* (3.24) *and* (3.25) *converges to the solution x^* of the linear system Eq.* (3.19) *in at most n steps.*

A simple interpretation of the properties of conjugate directions is the eigenvector of a certain associate ellipse of the matrix A. If the matrix A in Eq. (3.20) is diagonal, the contours of the function $\phi(\cdot)$ are ellipses whose axes are aligned with the coordinate directions, as illustrated in Figure 3.1. The minimizer of this function can be found by performing one-dimensional minimizations along the coordinate directions $e_1, e_2, ..., e_n$ in turn. When A is not diagonal, its contours are still elliptical, but they are usually no longer aligned with the coordinate directions. The strategy of successive minimization along these directions in turn no longer leads to the solution in n iterations. This phenomenon is illustrated in the two-dimensional example of Figure 3.2. A proper transform of the matrix A can, however, recover the nice behavior of Figure 3.1. Suppose we transform the problem by defining new variable $\hat{x}$ as

$$\hat{x} = S^{-1} x, \tag{3.26}$$

where S is the $n \times n$ matrix defined by

$$S = [p_0 \; p_1 \; \cdots \; p_{n-1}], \tag{3.27}$$

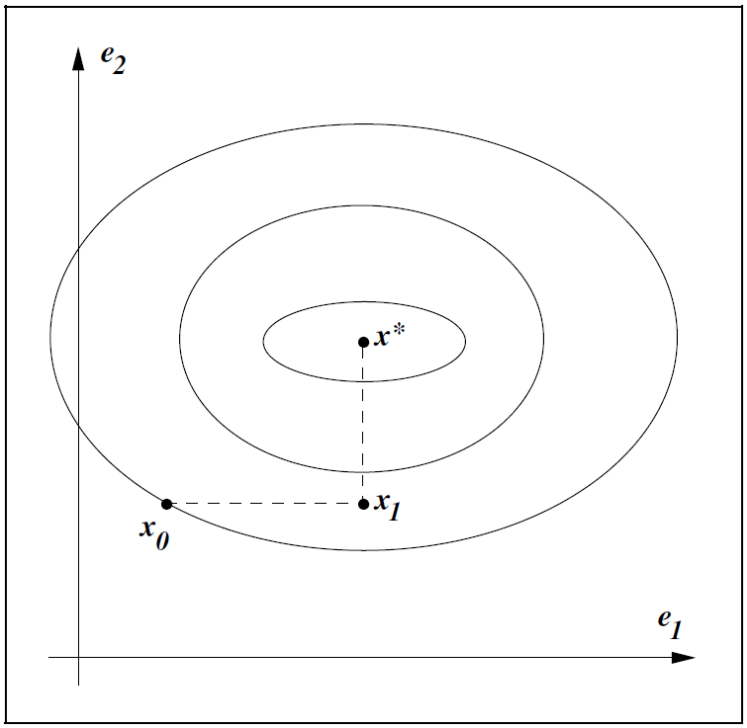


Figure 3.1 Successive minimizations along the coordinate directions [55].

where $\{p_0, p_1, ..., p_{n-1}\}$ is the set of conjugate directions with respect to A. The quadratic ϕ defined by Eq. (3.20) now becomes

$$\hat{\phi}(\hat{x}) \equiv \phi(S\hat{x}) = \frac{1}{2}\hat{x}(S^T AS)\hat{x} - (S^T b)^T \hat{x}. \tag{3.28}$$

By the conjugacy property Eq. (3.23), the matrix $S^T AS$ is diagonal, so the minimizing value of $\hat{\phi}$ can be found by performing n one-dimensional minimizations along the coordinate directions of $\hat{x}$. Because of the relation Eq. (3.26), however, the ith coordinate direction in $\hat{x}$-space corresponds to the direction p_i in x-space. Hence, the coordinate search strategy applied to $\hat{\phi}$ is equivalent to the conjugate direction algorithm (3.24) and (3.25).

In the quadratic function's point of view in Eq. (3.20), the conjugate direction is related to the Hessian matrix. When the Hessian matrix is diagonal, each coordinate minimization correctly determines one of the components of the

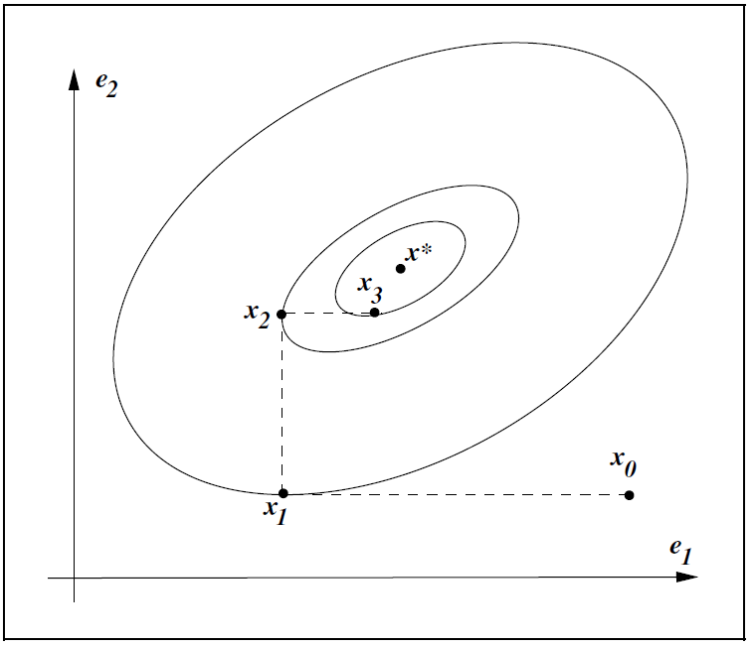


Figure 3.2 Successive minimizations along the coordinate directions for a general convex quadratic [55].

solution x^*. This can be extended for the general case of the Hessian matrix of the quadratic is not necessarily diagonal [55]:

Theorem 3.2.2. *Let $x_0 \in \mathbb{R}^n$ be any starting point and suppose that the sequence $\{x_k\}$ is generated by the conjugate direction algorithm* (3.24) *and* (3.25). *Then*

$$r_k^T p_i = 0, \quad \text{for } i = 0, 1, ..., k-1, \tag{3.29}$$

and x_k is the minimizer of $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ over the set

$$\{x | x = x_0 + \text{span}\{p_0, p_1, ..., p_{k-1}\}\}. \tag{3.30}$$

3.2.2 The Conjugate Gradient Method

The conjugate gradient method is a conjugate direction method with a very special property. In the generation of conjugate vectors, the conjugate gradient

method can compute a new vector p_k by using only the previous vector p_{k-1}. It does not need to know all the previous elements $p_0, p_1, ..., p_{k-2}$ of the conjugate set; p_k is automatically conjugate to these vectors. This remarkable property implies that the method requires little storage and computation.

In the conjugate gradient method, each direction p_k is chosen to be a linear combination of the negative residual $-r_k$ and the previous direction p_{k-1}:

$$p_k = -r_k + \beta p_{k-1}, \tag{3.31}$$

where the scalar β_k is to be determined by the requirement that p_{k-1} and p_k must be conjugate with respect to A. By premultiplying Eq. (3.31) by $p_{k-1}^T A$ and imposing the condition $p_{k-1}^T A p_k = 0$, we find that

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}. \tag{3.32}$$

The first search direction p_0 is chosen to be the steepest descent direction at the initial point x_0. As in the general conjugate direction method, successive one-dimensional minimizations are performed along each of the search directions. The complete procedure is expressed as the algorithm 1.

A slightly more economical form of the conjugate gradient method can be derived. For this derivation, additional properties are required, which come from the following theorem [55]:

Theorem 3.2.3. *Suppose that the k^{th} iterate generated by the conjugate gradient method is not the solution point x^*. The following four properties hold:*

$$r_k^T r_i = 0, \quad \text{for } i = 0, 1, ..., k-1, \tag{3.39}$$

$$\text{span}\{r_0, r_1, ..., r_k\} = \text{span}\{r_0, Ar_0, ..., A^k r_0\}. \tag{3.40}$$

$$\text{span}\{p_0, p_1, ..., p_k\} = \text{span}\{r_0, Ar_0, ..., A^k r_0\}. \tag{3.41}$$

$$p_k^T A p_i = 0, \quad \text{for } i = 0, 1, ..., k-1, \tag{3.42}$$

Therefore, the sequence $\{x_k\}$ converges to x^* in at most n steps.

Algorithm 1: The Conjugate Gradient Method - Preliminary Version

Data: A, x_0, b

Result: optimal x

begin

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$ **do**

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k} \tag{3.33}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k \tag{3.34}$$

$$r_{k+1} \leftarrow Ax_{k+1} - b \tag{3.35}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k} \tag{3.36}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k \tag{3.37}$$

$$k \leftarrow k + 1 \tag{3.38}$$

end

end

Using the results of Theorems 3.2.2 and 3.2.3, first, we can use Eq. (3.31) and (3.29) to replace the formula (3.46) for α_k by

$$\alpha_k \leftarrow -\frac{r_k^T r_k}{p_k^T A p_k}. \tag{3.43}$$

Second, it is easily verified from the relations Eq. (3.22) and (3.24):

$$r_{k+1} = r_k + \alpha_k A p_k. \tag{3.44}$$

From Eq. (3.44), we have that $\alpha_k A p_k = r_{k+1} - r_k$. By applying Eq. (3.29) and (3.39), we can simplify the formula for β_{k+1} to

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \tag{3.45}$$

By using these formulae together with Eq. (3.44), we obtain the standard form of the conjugate gradient method in algorithm 2.

3.3 The Singular Value Thresholding Algorithm

In mathematics, low-rank approximation is a minimization problem, in which the cost function measures the fit between a given matrix and an approximating matrix, subject to a constraint that the approximating matrix has reduced rank. The problem is used for mathematical modeling and data compression. The rank constraint is related to a constraint on the complexity of a model that fits the data. The singular value decomposition is a representative method to perform the low-rank approximation. The approximation is achieved using the singular value decomposition and the soft-thresholding operator.

Consider the singular value decomposition (SVD) of a matrix $X \in \mathbb{R}^{n_1 \times n_2}$ of rank r

$$X = U \Sigma V^*, \quad \Sigma = \mathrm{diag}(\{\sigma_i\}_{1 \le i \le r}), \tag{3.52}$$

where U and V are, respectively, $n_1 \times r$ and $n_2 \times r$ matrices with orthonormal columns, and the singular values σ_i are positive. For each $\tau \ge 0$, the soft-thresholding operator $\mathcal{S}_\tau$ is defined as [13]

$$\mathcal{S}_\tau(X) := U \mathcal{S}_\tau(\Sigma) V^*, \quad \mathcal{S}_\tau(\Sigma) = \mathrm{diag}(\{\sigma_i - \tau\}_+), \tag{3.53}$$

Algorithm 2: The Conjugate Gradient Method - Standard Version

Data: A, x_0, b

Result: optimal x

begin

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$ **do**

$$\alpha_k \leftarrow -\frac{r_k^T r_k}{p_k^T A p_k} \tag{3.46}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k \tag{3.47}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k \tag{3.48}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \tag{3.49}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k \tag{3.50}$$

$$k \leftarrow k + 1 \tag{3.51}$$

end

end

where t_+ is the positive part of t, *i.e.*, $t_+ = \max(0, t)$. This operator applies a soft-thresholding rule to the singular values of X, effectively shrinking these toward zero. This is the reason why this transformation is also referred as the *singular value shrinkage* operator. In some sense, this shrinkage operator is a straightforward extension of the soft-thresholding rule for scalars and vectors. If many of the singular values of X are below the threshold τ, the rank of $\mathcal{S}_\tau(X)$ may be considerably lower than that of X. This is like the soft-thresholding rule applied to vectors, which leads to sparser outputs whenever some entries of the input are below threshold. The singular value thresholding operator is the proximity operator associated with the nuclear norm. Details about the proximity operator can be found in, e.g., [33]. Using the singular value thresholding operator, the low-rank approximation is achieved. Cai *et al.* [13] showed that the following property of the *singular value shrinkage operator*.

Theorem 3.3.1. *For each $\tau \geq 0$ and $Y \in \mathbb{R}^{n_1 \times n_2}$, the singular value shrinkage operator obeys*

$$\mathcal{S}_\tau(Y) = \arg\min_X \frac{1}{2}\|X - Y\|_F^2 + \tau\|X\|^*. \tag{3.54}$$

When each column of the matrix X is an image data, this low-rank approximation gives the low-rank image of a set of images. We use this result for our nonlocal low-rank approximation of the image deblurring problem.

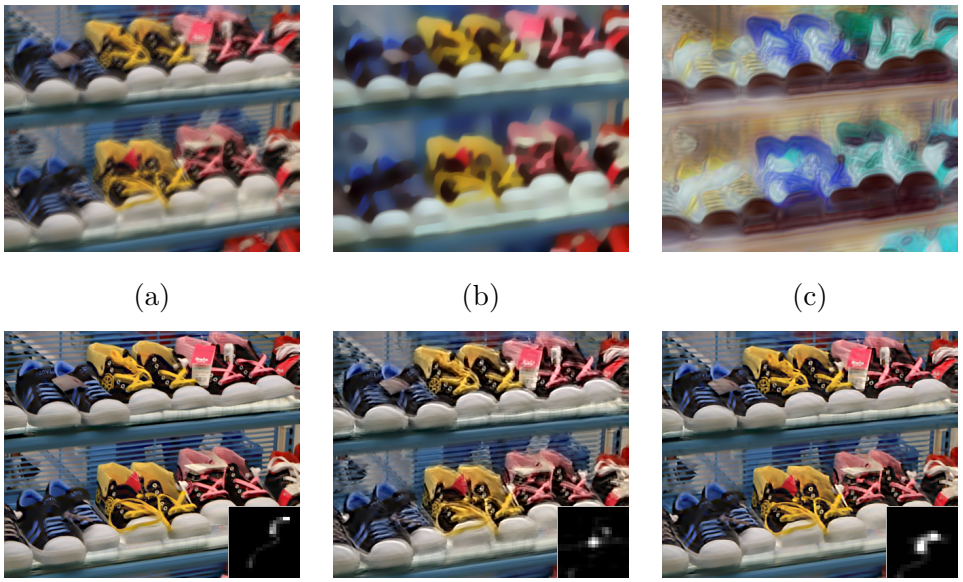
Chapter 4

Extracting Salient Structures

Edges in blurred images are important cues for restoration of latent images because they give information on motion of a camera, which can be used to infer a blur-kernel. On the other hand, flat and smooth regions of images give no information about the motion. Therefore, extracting reliable structures is critical to blind image deblurring especially for the kernel estimation. Different extractions of structures lead to different deblurred results. To obtain more reliable structures, we use a method of decomposition of structure and texture. In our approach, the main image structure is firstly extracted. A shock filter is then applied to get the enhanced structure. Finally, some salient edges with large pixel values are selected for the kernel estimation. The detail of this process is explained in the following sections.

4.1 Structure-Texture Decomposition with Uniform Edge Map

We extract salient edges from structures obtained by the structure-texture decomposition method [67]. Given an image I, the structure part of it is obtained



(a) (b) (c)

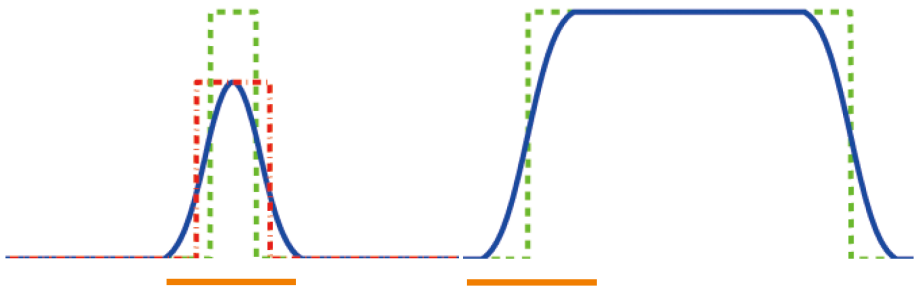
(d) (e) (f)

Figure 4.1 Example of structure-texture decomposition. (a) Blurred image. (b) Structure. (c) Texture. (d) Ground truth. (e) Without salient structure. (f) With salient structure.

by minimizing the following optimization problem [58]:

$$\min_{I_S} \|\nabla I_S\|_2 + \frac{1}{2\theta}\|I_S - I\|_2^2, \tag{4.1}$$

where θ is an adjustable parameter. The solution of this optimization gives a decomposition of the image I into the structure component I_S, shown in Figure 4.1 (b), and the texture components $I_T = I - I_S$, shown in Figure 4.1 (c). The structure component I_S contains the major objects in the image; on the other hand, I_T includes fine-scale details and noise. This structure is the basis of the salient edges.



(a) (b)

Figure 4.2 Ambiguity in motion deblurring [81].

4.2 Structure-Texture Decomposition with Adaptive Edge Map

The model (4.1) can still be improved by using adaptive parameter. The accuracy of a kernel estimate can be greatly improved by model (4.1); however, Pan *et al.* [58] showed that this model could lead to stair-casing effect in smooth area. This effect would distort gradient and cause inaccurate estimation of kernel. A simple way to mitigate this effect is to adjust the value of θ to be large in the smooth areas, and small near the edges.

Xu *et al.* [81] argued the ambiguity in restoring edges from motion blur and proposed a criterion for selecting informative edges. According to the works [68, 36, 19, 45], insignificant edges make kernel estimation vulnerable to noise. However, salient edges do not always improve kernel estimation. If the scale of an object is smaller than that of the blur kernel, the edge information could damage kernel estimation. For example, consider Figure 4.2, in which two step signal in (a) and (b) are blurred with a large kernel. The observed blur signals are shown in blue. Because the left signal is horizontally narrow, its height is lowered by blur process in (a), yielding ambiguity in the latent signal restoration. When sparse prior on the gradient map of the latent image is imposed, the red dashed line will be favored by motion deblur methods [31, 68] in computing

the unblurred signal because this version presents smaller gradient magnitudes. Moreover, the red signal preserves the total variation better than the green one. Therefore, it is also a more appropriate solution for the group of methods using sharp edge prediction This example shows that if blur significantly changes magnitude of image structures, the corresponding edge information could mistake kernel estimation. It also shows that the importance of edges is different.

We adjust the importance of each edge by using the criterion $r(\mathbf{x})$ suggested by Xu *et al.* [81], which is computed as

$$r(\mathbf{x}) = \frac{\| \sum_{\mathbf{y} \in N_h(\mathbf{x})} \nabla B(\mathbf{y}) \|_2}{\sum_{\mathbf{y} \in N_h(\mathbf{x})} \|\nabla B(\mathbf{y})\|_2 + 0.5}, \tag{4.2}$$

where B is a blurred image, and $N_h(\mathbf{x})$ is an $h \times h$ window centered at pixel $\mathbf{x}$. The constant 0.5 is to prevent producing a large r in flat regions. This criterion can be used for adjusting θ of the model (4.1). By adopting that criterion, the main structure of an image I can be obtained by solving the following optimization [58]:

$$\min_{I_S} \|\nabla I_S\|_2 + \frac{1}{2\theta\omega(\mathbf{x})}\|I_S - I\|_2^2, \tag{4.3}$$

where $\omega(\mathbf{x}) = \exp(-\|r(\mathbf{x})\|^{0.8})$. If a local region is flat, then the associated r has a small value. On the other hand, if a local region contains strong image structures, the associated r has a large value. However, model (4.3) keeps the similar advantages to those of [81] due to the adaptive weight $\omega(\mathbf{x})$. Also, this imposes a strong penalty to those areas where narrow strips are contained.

The validity of model (4.3) is demonstrated in Figure 4.3. As shown in Figure 4.3 (a), the blurred image contains some complex structures, which may have detrimental effects on kernel estimation. Due to adopting the adaptive weight $\omega(\mathbf{x})$ in model (4.3), kernel estimation result by model (4.3) is better than that by model (4.1). When compared with ∇S map shown in Figure 4.3 (d)-(e) and Figure 4.3 (g)-(i), both models (4.1) and (4.3) are able to select main structures of an image, but model (4.3) can retain major structures with less details for kernel estimation.

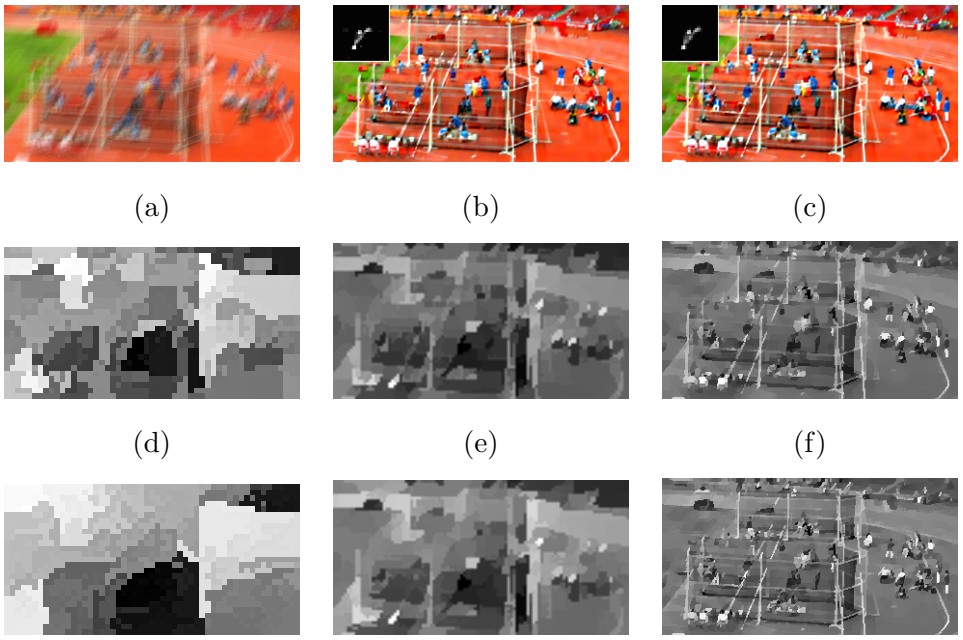


Figure 4.3 Comparison of results using different models. (a) Blurred image. (b) Result by model (4.3). (c) Result by model (4.1). (d)-(f) Salient structure of a interim image by model (4.3). (g)-(i) Salient structure of a interim image by model (4.1).

4.3 Enhancing Structures and Producing Salient Edges

The structure obtained by model (4.3) can still have artifacts and weak edges near strong edges. To mitigate the possible adverse effect of salient edges, we apply a gradient selection method after computing I_S. A shock filter is an effective tool for enhancing image features, which can recover sharp edges from blurred step signals [57]. Therefore, using the shock filter [57], we compute the enhanced structure $\tilde{I}_S$ as

$$\frac{\partial \tilde{I}_S}{\partial t} = -\text{sign}(\Delta I_S \|\nabla I_S\|_2), \tag{4.4}$$

where $\Delta I_S = I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}$. Finally, we compute salient edges ∇S as

$$\nabla S = \nabla \tilde{I}_S \odot H(\nabla \tilde{I}_S, t), \tag{4.5}$$

where $\odot$ is the element-wise multiplication, t is a threshold, and $H(\nabla \tilde{I}_S, t)$ is the unit binary mask function which is defined as

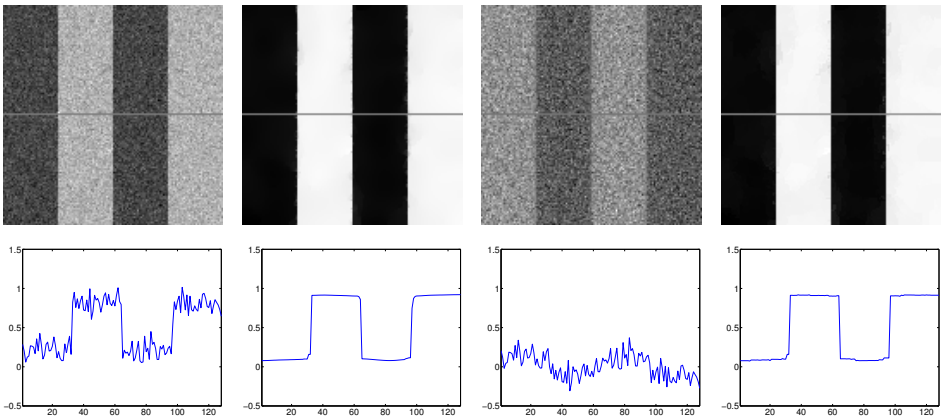
$$H(\nabla \tilde{I}_S, t) = \begin{cases} 1, & \|\nabla \tilde{I}_S\|_2 \geq t, \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

The application of Eq. (4.5) eliminates some noise in the $\|\nabla \tilde{I}_S\|_2$; thus, only the salient edges with large values influence the kernel estimation.

Kernel estimation could be unreliable when a few salient edges are available for estimation. To solve this problem, following strategies are adopted. First, the initial values of t is adaptively set according to the method of [19]. Specifically, t is set to guarantee that at least $\frac{1}{2}\sqrt{N_I N_k}$ pixels participate in kernel estimation, where N_I and N_k denote the total number of pixels in the input image and the kernel, respectively. Then, as the iteration goes in the deblurring process, the values of θ and t are gradually decreased at each iteration to include more edges for kernel estimation according to [81]. This strategy allows subtle structures to participate in the kernel refinement. As can be seen in Figure 4.3 (d)-(i), more sharp edges are included for kernel estimation as the iteration goes.

4.4 Analysis on the Method of Extracting Salient Edges

We analyzed a 2D signal to provide insight on how our method can improve kernel estimation, to better understand the method of extracting salient edges. The signal in Figure 4.4 (a) can be decomposed into the structure component in Figure 4.4 (b) and texture component in Figure 4.4 (c) by using model (4.5). For the structure component, a shock filter in Eq. (4.4) is also applied to get a sharp signal in Figure 4.4 (d) that is more similar to the step signal. Step signal usually succeeds in the kernel estimation, which has been proved by many



(a) (b) (c) (d)

Figure 4.4 Analysis of the structure-texture decomposition.

previous works [45, 36]. In contrast, the texture component usually fails in the kernel estimation as seen in previous section [81]. The noise contained in texture component damages the kernel estimation. Since the size of texture component is relatively small, its height is reduced by blurring. Therefore, recovering the sharp texture component from the blurred image is a very difficult problem. As a result, a correct kernel estimate is hard to be obtained by texture component.

Chapter 5

Blind Image Deblurring using Nonlocal Patches

In this chapter, we explain our approach to blind image deblurring. Our approach intensively uses of nonlocal information. For the kernel estimation, we make use of nonlocal patches obtained by similarity weighted by current estimate of kernel. Using these kernel-guided nonlocal patches, we impose regularization that nonlocal patches would produce the similar values by convolution, which improves the kernel estimation. During this kernel estimation, we make use of the salient structures. For the latent image estimation, we also utilize nonlocal patches from which nonlocal low-rank image is constructed. Using this nonlocal low-rank image, we impose regularization that latent image is similar to that nonlocal low-rank image. The overall process is illustrated in Figure 5.1.

Blind image deblurring is originally a joint optimization problem of obtaining a latent image I and a blur kernel k as formulated in Eq. (1.5). In practice, since this joint optimization is a highly nonlinear problem, the alternating approach is usually applied, which solves the following two optimizations

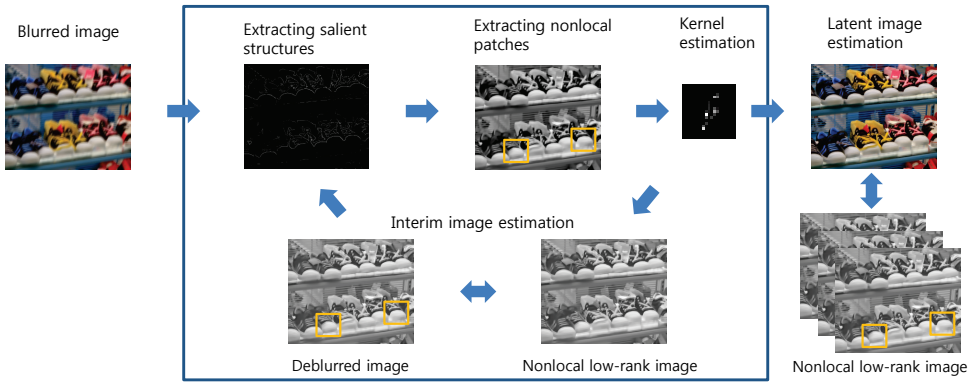


Figure 5.1 Overall process of our approach.

separately by fixing each other:

$$\min_{I} \|B - k \otimes I\|_2 + \lambda J(I) \tag{5.1}$$

and

$$\min_{k} \|B - k \otimes I\|_2 + \gamma G(k). \tag{5.2}$$

In our approach, we also adopt this alternating optimization. As shown in Figure 5.1, we firstly estimate the blur kernel and the interim image separately. After the kernel estimation is completed, we estimate the latent image using that estimated kernel. In each image estimation, the estimate of image is iteratively updated together with the estimation of a nonlocal low-rank image.

5.1 Estimating a Blur-kernel using Kernel-guided Nonlocal Patches

In this section, we explain the method of kernel estimation. Since the convolution and derivative are linear operators, it holds that $\nabla B = k \otimes \nabla I$ if $B = k \otimes I$. Using this equivalence, the kernel estimation can be modeled from Eq. (5.2) as

$$\min_{k} \|\nabla B - k \otimes \nabla I\|_2 + \gamma G(k). \tag{5.3}$$

In this model, ∇I means edges in the image I. If we use the salient edges ∇S explained in previous chapter, this model is formulated as

$$\min_k \|\nabla B - k \otimes \nabla S\|_2 + \gamma G(k). \tag{5.4}$$

To this formulation, we add various priors to obtain more reliable kernel.

5.1.1 Sparse Prior

The motion blur of images is caused by the camera shake during the exposure. The blur kernel describes the path of this shake. This path usually looks like a narrow curve. Most literature assumes that distributions of blur kernels can be modeled by a hyper-Laplacian. Under this assumption, the model for kernel estimation using the salient edges ∇S is

$$\begin{aligned} &\min_k \|\nabla B - k \otimes \nabla S\|_2^2 + \gamma \|k\|_\alpha^\alpha, \\ &\quad \text{s.t. } k \succeq 0,\ \|k\|_1 = 1, \end{aligned} \tag{5.5}$$

where $0 < \alpha < 1$. The value of α is usually set as values from 0.5 to 0.8.

Although the sparsity prior is effectively modeled by this formulation, the continuity of a blur kernel is not ensured. For this limitation, noisy kernel estimates are sometimes induced. The imperfect salient edges ∇S can also cause noise in the kernel estimates. Figures. 5.2 (a) and (b) is the unsuccessful estimates of deblurred images due to the incorrect and noisy kernel estimates.

5.1.2 Continuous Prior

The noise in kernel estimates is mitigated by additional prior. Since a camera shake is a continuous movement, the shape of blur kernel is also continuous. To preserve the continuity of kernel, we apply a regularization by the following spatial term as in [58]:

$$C(k) = \big\{(x, y) \;\big|\; |\partial_x k(x, y)| + |\partial_y k(x, y)| \neq 0\big\}, \tag{5.6}$$

(a) (b) (c)

Figure 5.2 Comparison of results of blur-kernel estimates. (a) Levin *et al.* [46]. (b) Perrone *et al.* [61]. (c) Ours.

where $C(k)$ counts the number of pixels whose gradients are non-zero. This helps the kernel estimation to keep the continuous structure of kernel effectively and to remove some noise. If this spatial term is applied, the kernel estimation problem becomes

$$\min_{k} \|\nabla B - k \otimes \nabla S\|_2^2 + \gamma \|k\|_\alpha^\alpha + \mu C(k) \quad \text{s.t. } k \succeq 0,\ \|k\|_1 = 1, \tag{5.7}$$

where the parameter μ controls the smoothness of k. Model (5.7) is robust to noise and can preserve both sparsity and continuity of kernel.

5.1.3 Nonlocal Prior by Kernel-guided Nonlocal Patches

We additionally introduce a nonlocal prior such that patches after blurring are similar if their latent patches are similar. For the application of this prior, we firstly select nonlocal similar patches based on the similarity weighted by the estimating kernel. Ordinary nonlocal patches could have quite different values under a convolution operation because the convolution amplifies or reduces the effect of each element depending on the kernel. If the kernel is correct, the nonlocal patches by kernel weight result in the similar values after convolution.

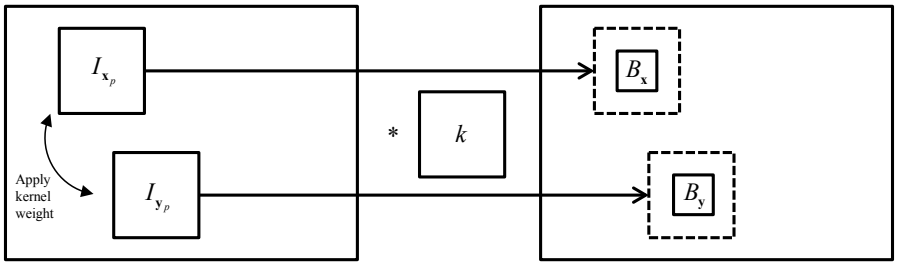


Figure 5.3 Illustration of the nonlocal operation.

Therefore, this prior knowledge helps the estimation of the correct kernel.

We propose the nonlocal regularization as

$$J_{NLK}(k; I, B) = \sum_{\{\mathbf{x}\in\Omega\}} \sum_{\{\mathbf{y}\in NL_K(\mathbf{x})\}} \|(B_{\mathbf{y}} - B_{\mathbf{x}}) - ((I_{\mathbf{y}_p} - I_{\mathbf{x}_p}) \otimes k)_c\|_2^2, \tag{5.8}$$

where Ω is a set of pixel locations that are counted for the nonlocal cost, $NL_K(\mathbf{x})$ is a set of nonlocal similar locations to $\mathbf{x}$, $\mathbf{x}_p$ is the patch whose center is $\mathbf{x}$, $\mathbf{y}_p$ is the nonlocal similar patch to $\mathbf{x}_p$ whose center is $\mathbf{y}$. Under this notation, $I_{\mathbf{x}_p}$ means the patch on I whose center is $\mathbf{x}$, and the $B_{\mathbf{x}}$ is the pixel $\mathbf{x}$ on B. The subscript c means extracting the pixel of the center of the patch. We extracted only the center of the patch to avoid the boundary problem of convolution. In Figure 5.3, the simple nonlocal operation is illustrated for example. The set of nonlocal similar locations is defined as

$$NL_K(\mathbf{x}) := \{\ \mathbf{y} \mid d(I_{\mathbf{x}_p}, I_{\mathbf{y}_p}) \leq K\text{-}dist(I_{\mathbf{x}_p}), \mathbf{y} \in \Omega\} \tag{5.9}$$

where $d(I_{\mathbf{x}_p}, I_{\mathbf{y}_p})$ is the distance between patches, and $K\text{-}dist(I_{\mathbf{x}_p})$ is the distance of the K^{th}-nearest patch from the patch $I_{\mathbf{x}_p}$. The distance measure is given by

$$d(I_{\mathbf{x}_p}, I_{\mathbf{y}_p}) = \| \ (I_{\mathbf{x}_p} \odot (k_- + \frac{1}{h \cdot w})) - (I_{\mathbf{y}_p} \odot (k_- + \frac{1}{h \cdot w})) \ \|_F, \tag{5.10}$$

where k_- is the flipped version of k, h and w are the height and the width of the kernel, respectively, $\odot$ is the element-wise multiplication, and $\|\cdot\|_F$ means the Frobenius norm. Here, k_- is applied because convolution operation measures correlation between an image patch and the flipped version of the kernel. The operation $\odot$ applies the kernel weight for each element of a patch, and $\frac{1}{h \cdot w}$ applies a minimal contribution for every element of the patch.

To effectively impose the nonlocal regularization, some details should be considered. First, we use the size of nonlocal patch as the same to the size of kernel. Since J_{NLK} is computed from the convolved value of the patch, the size of patch should be at least the size of the kernel. Nonetheless, larger size patches may deteriorate the selection of similar patches. Therefore, we use the same size for the patch and the kernel and use only the center of the convolved patch for regularization. Second, we extract the candidate patches only from the salient edges for the same reason of avoiding the noise effect. For this candidate locations, Ω in Eq. (5.8) consists of the locations at which the value of ∇S is non-zero.

Based on the above considerations, our final kernel estimation model is defined as

$$\begin{aligned} \min_k \|\nabla B - k \otimes \nabla S\|_2^2 + \gamma \|k\|_\alpha^\alpha + \mu C(k) \\ + \lambda J_{NLK}(k; I, B), \\ \text{s.t. } k \succeq 0,\ \|k\|_1 = 1, \end{aligned} \tag{5.11}$$

where γ, μ, and λ control the contribution of each term. Model (5.11) is difficult to be minimized directly because of the discrete counting metric. Therefore, we approximate it by alternately minimizing

$$\begin{aligned} \min_k \|\nabla B - k \otimes \nabla S\|_2^2 + \gamma \|k\|_\alpha^\alpha + \lambda J_{NLK}(k; I, B), \\ \text{s.t. } k \succeq 0,\ \|k\|_1 = 1, \end{aligned} \tag{5.12}$$

and

$$\min_{\hat{k}} \|\hat{k} - k\|_2^2 + \mu C(\hat{k}). \tag{5.13}$$

Model (5.12) can be optimized by using the constrained iterative reweighted least square (IRLS) method [43]. For model (5.13), we employ the alternating optimization method in [82].

5.2 Estimating an Interim Image using Nonlocal Low-rank Images

Eq. (5.1) is the basis of model of the interim image estimation. The interim image is used as data for kernel estimation. The small edges of the interim image could mislead the kernel estimation. Therefore, we focus on the restoration of sharp edges from the blurred image for the interim image estimation. TV model is a good candidate for this because it is effective to restore strong edges while removing small edges:

$$\min_{I} \|B - k \otimes I\|_2^2 + \beta \|\nabla I\|_1. \tag{5.14}$$

This model can be efficiently solved by IRLS as well. When solving the least square problem in the inner IRLS system, we use the conjugate gradient (CG) method. In addition to this TV model, we add a nonlocal regularization to reduce the noise effect, which is explained the followings.

5.2.1 Nonlocal Low-rank Prior

Sparse representation and nonlocal self-similarity have been successfully used for image restoration problems. The basic assumption underlying this approach is that self-similarity is abundant in signals of our interest. Under the assumption that these image patches have similar structures, a data matrix consisting of a group of those patches has a low-rank property. Using these nonlocal

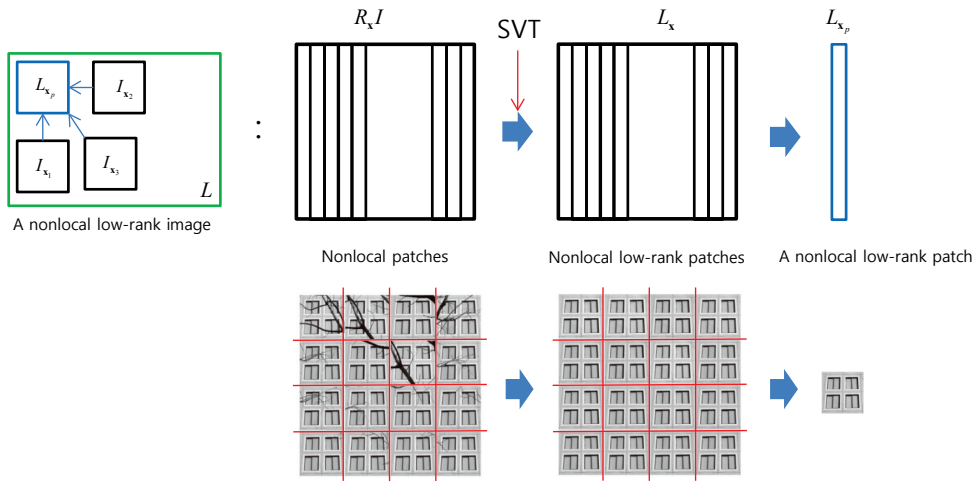


Figure 5.4 Illustration of the nonlocal low-rank operation.

patches, we can construct nonlocal low-rank images. Therefore, we propose a model imposing the low-rank images as a prior on the latent image. To obtain the nonlocal low-rank images, we firstly group patches by their similarity and then perform the low-rank approximation for each group. By utilizing this property, we can reduce the noise effect to estimate the latent image.

The assumption of self-similarity implies that a sufficient number of similar patches can be found for any exemplar patch of size $\sqrt{n} \times \sqrt{n}$ at position $\mathbf{x}$ denoted by $\mathbf{x}_p$. For each exemplar patch $\mathbf{x}_p$ in an image, we perform a variant of K-nearest-neighbor search within a local window (e.q., 40×40):

$$G_{\mathbf{x}} = \{\ \mathbf{y} \mid \|\mathbf{x}_p - \mathbf{y}_p\| < T\ \}, \tag{5.15}$$

where T is a pre-defined threshold, and $G_{\mathbf{x}}$ denotes the collection of positions corresponding to those similar patches.

Under the assumption that image patches whose positions are in $G_{\mathbf{x}}$ have similar structures, we obtain a data matrix $L_{\mathbf{x}}$ with a low-rank property and get a nonlocal low-rank patch $L_{\mathbf{x}_p}$. We firstly construct a data matrix $X_{\mathbf{x}}$ whose columns are composed of patches constructed from $G_{\mathbf{x}}$. In practice, $X_{\mathbf{x}}$ may

be corrupted by some noise. We model the data matrix $X_{\mathbf{x}}$ as: $X_{\mathbf{x}} = L_{\mathbf{x}} + W_{\mathbf{x}}$, where $L_{\mathbf{x}}$ and $W_{\mathbf{x}}$ denote the low-rank matrix and the Gaussian noise matrix, respectively. Then the low-rank matrix $L_{\mathbf{x}}$ can be obtained by solving the following optimization problem:

$$L_{\mathbf{x}} = \underset{L_{\mathbf{x}}}{\operatorname{argmin}} \operatorname{rank}(L_{\mathbf{x}}), \quad \text{s.t. } \|X_{\mathbf{x}} - L_{\mathbf{x}}\|_F^2 \leq \sigma_w^2, \tag{5.16}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm and σ_w^2 denotes the variance of additive Gaussian noise. Finally, by averaging the columns, we can construct a nonlocal low-rank patch as shown in Figure 5.4, where the notation $R_{\mathbf{x}}I$ is equivalent to $X_{\mathbf{x}}$. In analogy, this corresponds to a single window image obtained from images of various but similar windows as shown in the figure.

This nonlocal low-rank approach was also applied to the works of [22, 59], Inspired by this, we combine the TV model and the nonlocal low-rank approach to guide the restoration of the interim image. The difference of our work is that we combine the low-rank property with the total variation regularization to keep the property in the model (5.14). Our deblurring model is defined as

$$\min_{I,L} \|B - k \otimes I\|_2^2 + \beta\|\nabla I\|_1 + \tau J_{NLR}(I, L), \tag{5.17}$$

where $J_{NLR}(I, L)$ is defined as

$$J_{NLR}(I, L) = \sum_{\mathbf{x}\in\Omega} \|R_{\mathbf{x}}I - L_{\mathbf{x}}\|_F^2 + \sigma\|L_{\mathbf{x}}\|_* \tag{5.18}$$

where $L_{\mathbf{x}}$ denotes the image patches with low-rank property and $R_{\mathbf{x}}I = [I_{\mathbf{x}_1}, I_{\mathbf{x}_2}, ..., I_{\mathbf{x}_N}] \in \mathbb{R}^{n\times N}$ is the matrix whose columns, $I_{\mathbf{x}_i}$, are the vectorized nonlocal similar patches to $I_{\mathbf{x}_p}$. The first term is the reconstruction error which ensures the recovered image should be consistent with the input image with respect to the estimated degradation model. The second term imposes the total variation regularization of I to be able to preserve sharp edges in the recovered latent image. The third term uses low-rank prior to provide a data authentic prior for the latent image, which also suppresses noise effect.

Minimizing model (5.17) involves simultaneously computing two variables. We employ the alternating minimization scheme to solve it.

Updating the interim image I: With L fixed, the problem for updating interim image I is done by

$$\min_{I} \|B - k \otimes I\|_2^2 + \beta\|\nabla I\|_1 + \tau \sum_{\mathbf{x}\in\Omega} \|R_{\mathbf{x}} I - L_{\mathbf{x}}\|_F^2. \tag{5.19}$$

We solve this optimization problem by using IRLS.

Updating the low-rank part L: With I fixed, the problem for updating L is

$$\min_{L} \tau \sum_{\mathbf{x}\in\Omega} \|R_{\mathbf{x}} I - L_{\mathbf{x}}\|_F^2 + \sigma\|L_{\mathbf{x}}\|_*. \tag{5.20}$$

This is a standard low-rank approximation problem [13]. Therefore, its closed form solution can be obtained by the singular value thresholding (SVT)

$$L_{\mathbf{x}} = U_{\mathbf{x}} \mathcal{S}_{\sigma/(2\tau)}[\Sigma_{\mathbf{x}}] V_{\mathbf{x}}^T \tag{5.21}$$

where $U_{\mathbf{x}}\Sigma_{\mathbf{x}}V_{\mathbf{x}}^T$ is the singular value decomposition (SVD) of $R_{\mathbf{x}}I$ and $\mathcal{S}_{\sigma/(2\tau)}[v]$ is defined as

$$\mathcal{S}_{\sigma/(2\tau)}[v] = \begin{cases} v - \sigma/(2\tau), & v > \sigma/(2\tau), \\ v + \sigma/(2\tau), & v < -\sigma/(2\tau), \\ 0, & \text{otherwise.} \end{cases} \tag{5.22}$$

The estimated low-rank image L is used in Eq. (5.19) again. This alternating and iterative estimation of I and L lead to the noiseless deblurred image.

5.3 Multiscale Implementation

To find solutions to large kernels, an excessive number of I and k updates may be required. To mitigate this problem, multiscale estimation of the kernel is performed using a coarse-to-fine pyramid of image resolutions. The levels of

pyramid is constructed by a size ratio of $\sqrt{2}/2$ between them. At first stage, the input blurry image is downsampled to the lowest levels, and then the kernel k and the interim image I are estimated. After that, the upsampled version of them is used as the initialization of the kernel and sharp image at the next finer level. The resizing operations are done by using bilinear interpolation. This procedure is described in Algorithm 3.

Algorithm 3: Multiscale implementation of the blur-kernel estimation

Data: Blur image B and the size of blur kernel

Result: Blur kernel k

begin

  for $i = 1$ **to** n **do**

    Downsample B according to the current image pyramid to get B_i;

    for $j = 1$ **to** m (m *iterations*) **do**

      Estimate kernel k;

      Estimate interim image I_i;

      $t \leftarrow t/1.1$;

      $\theta \leftarrow \theta/1.1$;

    end

    Upsample image I_i and set $B_{i+1} \leftarrow I_i$

  end

end

5.4 Latent Image Estimation

After the estimation of the kernel k is completed, the latent image I can be recovered using various non-blind deconvolution methods from blurred image B. Model (5.14) can be used for this, but it may lead to the stair-casing effect and destroy textures. According to several works [39, 45, 46], the hyper-Laplacian prior on images can model the natural image prior and it is easily solved by

IRLS. In addition, the latent image is more vulnerable to noise than the interim image. Therefore, we apply the hyper-Laplacian prior and employ the nonlocal low-rank image as a prior, which guide the latent image to be natural and less noisy. The latent image is obtained by solving

$$\min_{I,L} \|B - k * I\|_2^2 + \eta\|\nabla I\|_\alpha^\alpha + \tau J_{NLR}(I, L). \tag{5.23}$$

This model can also be alternately solved by the IRLS method and the SVT method as in models (5.19) and (5.20), respectively. The only difference is the norm of ∇I. In this latent image estimation, we alternately minimize the followings:

$$\min_{I} \|B - k \otimes I\|_2^2 + \beta\|\nabla I\|_\alpha^\alpha + \tau \sum_{\mathbf{x}\in\Omega} \|R_{\mathbf{x}} I - L_{\mathbf{x}}\|_F^2 \tag{5.24}$$

and

$$\min_{L} \tau \sum_{\mathbf{x}\in\Omega} \|R_{\mathbf{x}} I - L_{\mathbf{x}}\|_F^2 + \sigma\|L_{\mathbf{x}}\|_*. \tag{5.25}$$

The value of σ in Eq. (5.25) is dependent on the noise level of observed images. The optimization in Eq. (5.24) gives a latent image similar to the nonlocal low-rank image, which contains less noise. Therefore, the noise in observed blurry image can be removed.

Chapter 6

Experimental Results

In this chapter, we conduct experiments and compare the results with those of conventional methods quantitatively and qualitatively. For reliable quantitative evaluation, we made blurry images using their ground truth images and the ground truth kernels. The ground truth images and the kernels are shown in Figures 6.1 and 6.2, respectively. The blurry images are in Figures 6.3 and 6.4. We performed blind image deblurring to these images. Using the results and the ground truth, we computed values of various metrics to evaluate the quality of the results. The metrics include the sum of squared difference of kernels and images (SSD), the peak signal to noise ratio (PSNR), and the structural similarity index (SSIM). We also made noisy and blurry images and conducted experiments on them to evaluate the robustness to noise. The blurry and noisy images are in Figures 6.5 and 6.6. For these noisy and blurry images, the same evaluation was applied as in the case of the images with blur only. We also conducted experiments on the images without ground truth. For these images, we compared the visual quality of them qualitatively. We also computed and compared values of various metrics from the works on the no-reference image quality assessment [47, 53, 62], quantitatively. Also, we examined the effect of

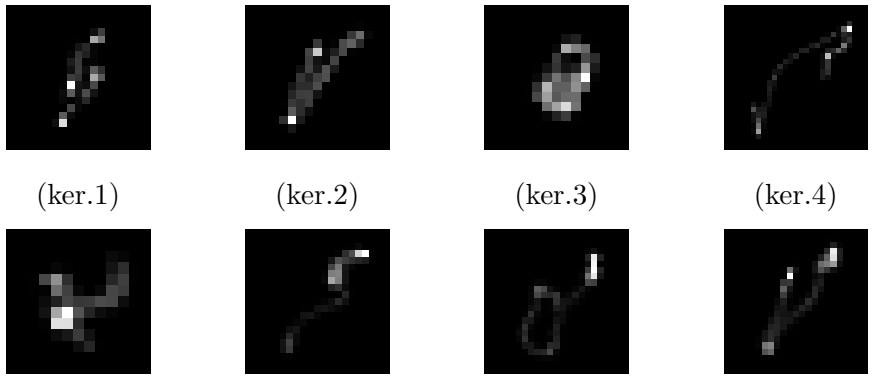
(im.1) (im.2) (im.3) (im.4)

(im.5) (im.6) (im.7) (im.8)

Figure 6.1 Ground truth images.

the denoising method as a preprocessing step and analyzed the effect of the size of nonlocal patches.

For the experiments, we set the parameters as in Table 6.1. We used different parameters for the images with blur only and the images with blur and noise. Most of the parameters are same for the both, but the parameters related to the low-rank prior are different. Those parameters are closely related to the noise level of images. For noisy images, the low-rank prior should be stronger than for noiseless images. Thus, the larger τ was applied to noisy images. σ is also closely related to the noise level of images. The larger σ takes fewer rank of images; therefore, the noisy components and small textures are well eliminated. On the contrary, the higher σ could induce too smooth image; therefore, we applied smaller σ to the latent image estimation than the interim image estimation for noiseless images. The size of nonlocal patches is also related to the noise level. If image contains noise, the similarity between patches could be lowered. We used larger size to find reliable similar patches by overcoming the effect of noise.



(ker.1) (ker.2) (ker.3) (ker.4)

(ker.5) (ker.6) (ker.7) (ker.8)

Figure 6.2 Ground truth kernels.

Table 6.1 Parameter setting.

Parameters	Images with blur only	Images with blur and noise
θ in Eq. (4.3)	1	1
γ in Eq. (5.11)	0.01	0.01
μ in Eq. (5.11)	0.01	0.01
λ in Eq. (5.11)	0.07	0.07
β in Eq. (5.17)	0.005	0.005
η in Eq. (5.23)	0.003	0.003
K in Eq. (5.9)	5	5
τ in Eq. (5.17)	0.1	2
σ in Eq. (5.18) for Eq. (5.17)	0.55	3
σ in Eq. (5.18) for Eq. (5.23)	0.4	3
Patch size for Eq. (5.18)	7×7	9×9

6.1 Images with Ground Truth

We evaluate the effectiveness of our method by comparing the quality of the estimates of the kernel and the latent image from our method against those from each individual method. If images have their ground truth images and kernels, we can utilize various reliable metrics.

Error Ratio curve: Levin *et al.* [45] noted that a wider kernel tends to result in larger deconvolution error even with the true kernel. They suggested a new metric measuring the deconvolution error, normalized by error by the true kernel. For the deconvolution, the same non-blind deconvolution method is applied:

$$\mathrm{ER}(k; k_g) = \frac{\|I_k - I_g\|_2}{\|I_{k_g} - I_g\|_2}, \tag{6.1}$$

where I_g denotes the ground-truth image. I_k and I_{k_g} denote the deconvolution results with the estimated kernel k and those with the ground-truth kernel k_g respectively. For evaluation, we computed this ER curve for each method. At each error ratio level in the ER curve, the success rate of the method represents the percentage of test images that has the error ratio below that error level.

SSD: To evaluate how well the estimation was attained, we also measure how similar the estimates of the kernel and the image are to the ground-truth kernel and image. Specifically, we measure the similarity using the sum of squared difference (SSD) between the two kernels or images.

$$\mathrm{SSD}(I; I_g) = \frac{1}{|I|} \sum_{p \in \Omega} |I(p) - I_g(p)|^2, \tag{6.2}$$

where Ω is the set of pixels, I denotes the estimated kernel or image, I_g denotes the ground-truth kernel or image, and $|I|$ denotes the number of elements in the kernel or the image.

PSNR and SSIM are the widely-used metrics in image processing and computer vision research to measure the image reconstruction quality. The PSNR is defined as

$$\mathrm{PSNR}(I_t, I_g) = 20 \cdot \log_{10}\left(\frac{\max(I_g)}{\sqrt{\mathrm{MSE}(I_t, I_g)}}\right), \tag{6.3}$$

where I_t is a test image, I_g is the ground-truth image, and MSE is given by

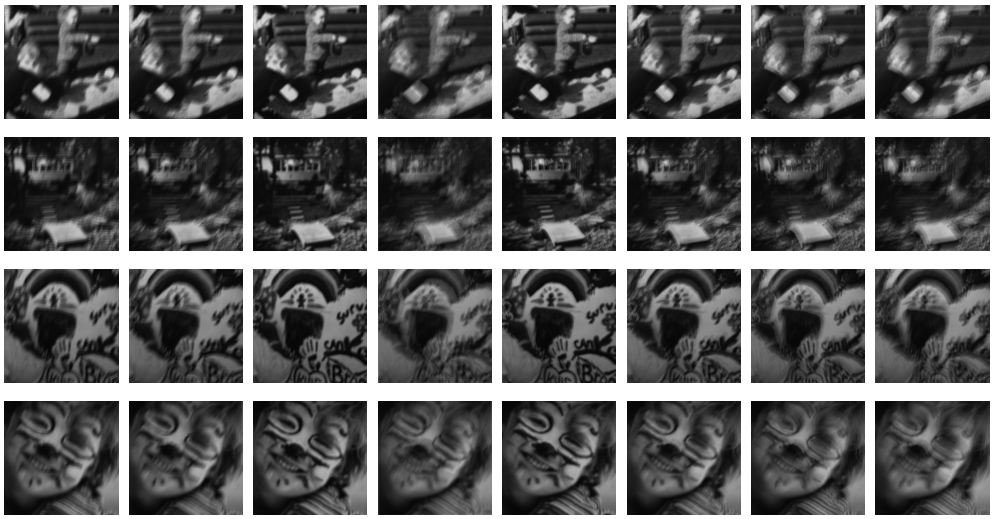
$$\mathrm{MSE}(I_t, I_g) = \frac{1}{H \cdot W} \sum_{p \in \Omega} |I_t(p) - I_g(p)|^2, \tag{6.4}$$

where H and W are the height and the width of the image, and Ω is the set of pixels. The SSIM is defined as

$$\mathrm{SSIM}(I_t, I_g) = \frac{(2\mu_{I_t}\mu_{I_g} + c_1)(2\sigma_{I_t,I_g} + c_2)}{(\mu_{I_t}^2 + \mu_{I_g}^2 + c_1)(\sigma_{I_t}^2 + \sigma_{I_g}^2 + c_2)}, \tag{6.5}$$

where μ_{I_t} is the average of I_t, μ_{I_g} is the average of I_g, $\sigma_{I_t}^2$ is the variance of I_t, $\sigma_{I_g}^2$ is the variance of I_g, σ_{I_t,I_g} is the covariance of I_t and I_g, and $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator. The L is the dynamic range of the pixel-values, $k_1 = 0.01$, $k_2 = 0.03$ by default. These two metrics directly measure the visual difference between the reconstruction results and the corresponding ground-truth images.

We conducted experiments using the 8 ground truth images in Figure 6.1 and the 8 ground truth kernels in Figure 6.2. First, we made up two 32 examples which are obtained from the ground truth images by the convolution of the ground truth kernels above. The 32 examples of the gray images in Figure 6.1 are listed in Figure 6.3, and the 32 examples of the color images in Figure 6.1 are listed in Figure 6.4. We denote them as DATA SET 1 and DATA SET 2, respectively. From the DATA SET 1 and the DATA SET 2, we estimated the kernels and the latent images using various blind deblurring methods. From the estimated kernels and images, we computed the error ratio (ER), the sum of squared difference (SSD), the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) to examine the image quality. Our method was



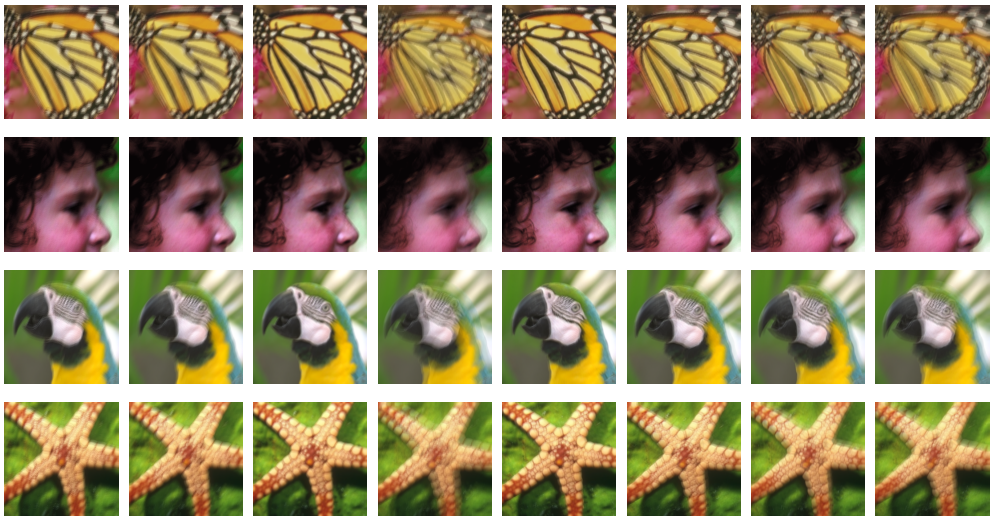
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.3 DATA SET 1: blurred images from im.1 - im.4. (a) By kernel 1. (b) By kernel 2. (c) By kernel 3. (d) By kernel 4. (e) By kernel 5. (f) By kernel 6. (g) By kernel 7. (h) By kernel 8.

compared with those of Levin *et al.* [46], Pan *et al.* [58], Zhong *et al.* [88], and Perrone *et al.* [61].

We also conducted experiments for examining the robustness to noise. We added Gaussian noise of 10 standard deviation to the DATA SET 1 and the DATA SET 2 when the intensity range is $[0, 255]$. The 32 examples of the gray images in Figure 6.1 are listed in Figure 6.5, and the 32 examples of the color images in Figure 6.1 are listed in Figure 6.6. We denote them as DATA SET 3 and DATA SET 4, respectively. From the DATA SET 3 and the DATA SET 4, we estimated the kernels and the latent images and computed ER, SSD, PSNR and SSIM.

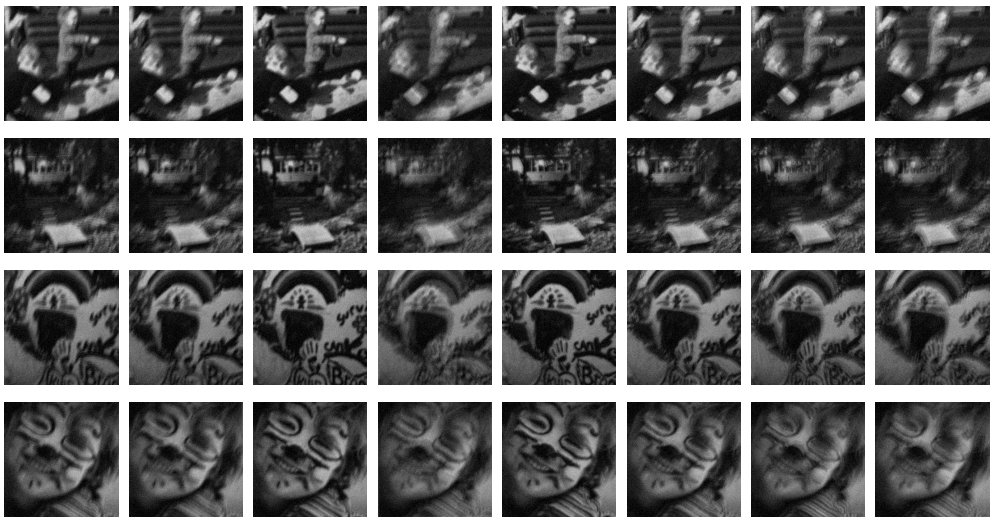
In Figures 6.7, 6.8, 6.9, and 6.10, we listed some of the estimated kernels and the images on the blurred images of DATA SETs 1, 2, 3, and 4, respectively. For these results, we computed ER and illustrated the ER curve in Figure 6.11.



(a) (b) (c) (d) (e) (f) (g) (h)

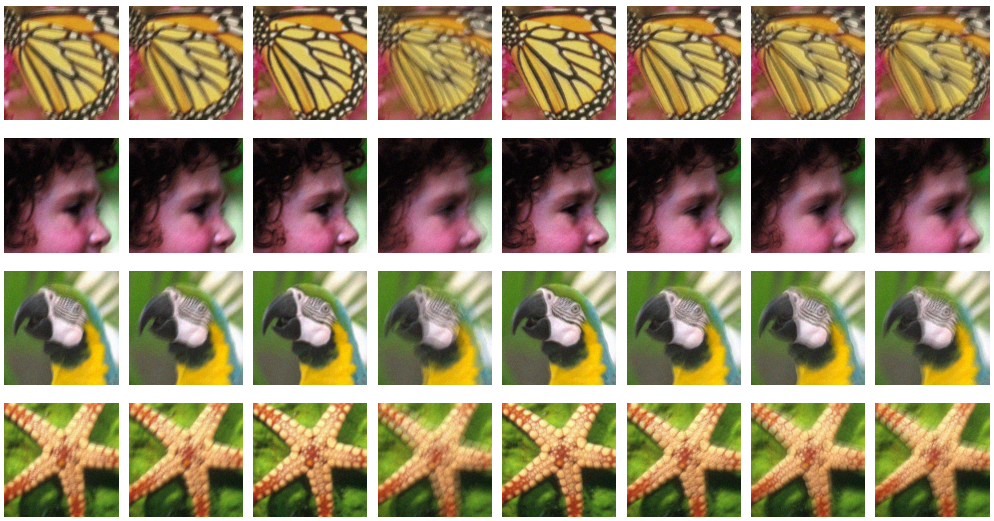
Figure 6.4 DATA SET 2: blurred images from im.5 - im.8. (a) By kernel 1. (b) By kernel 2. (c) By kernel 3. (d) By kernel 4. (e) By kernel 5. (f) By kernel 6. (g) By kernel 7. (h) By kernel 8.

As shown in Figure 6.11 (a), ours attained the highest success percentage for the smallest error ratio for DATA SET 1. Levin *et al.*'s result was competitive to ours. In Figure 6.11 (b), the ER curve of DATA SET 2 was illustrated. In this figure, the result of Levin *et al.* [46] attained the highest success percentage for the smallest error ratio, and ours attained the second. In this figure, ours attained the highest at the error ratio of 3, and the overall performance of ours is the highest or competitive to that of Levin *et al.* [46]. For these noisy and blurry images, the ER performance of each method was dramatically lowered. For DATA SET 3 and 4, we also computed ER and illustrated the ER curve in Figure 6.11 (c) and (d). As shown in Figure 6.11 (c) and (d), only the method of Zhong *et al.*[88] and ours attained successful results. Compared with the result of Zhong *et al.* [88], ours is highly outperformed.



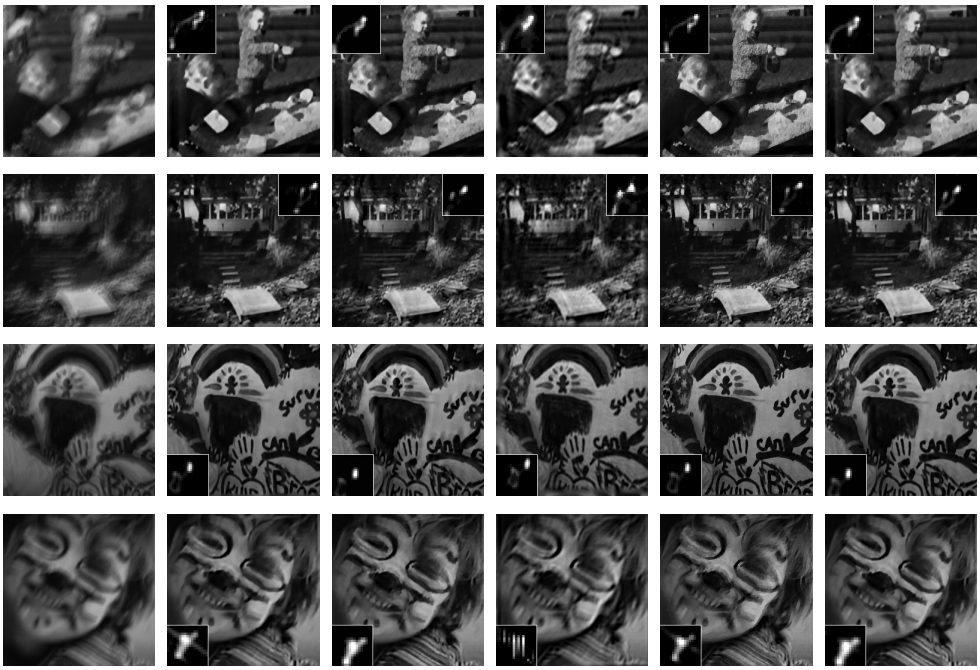
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.5 DATA SET 3: noisy and blurred images from im.1 - im.4. (a) By kernel 1. (b) By kernel 2. (c) By kernel 3. (d) By kernel 4. (e) By kernel 5. (f) By kernel 6. (g) By kernel 7. (h) By kernel 8.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.6 DATA SET 4: noisy and blurred images from im.5 - im.8. (a) By kernel 1. (b) By kernel 2. (c) By kernel 3. (d) By kernel 4. (e) By kernel 5. (f) By kernel 6. (g) By kernel 7. (h) By kernel 8.



(a) (b) (c) (d) (e) (f)

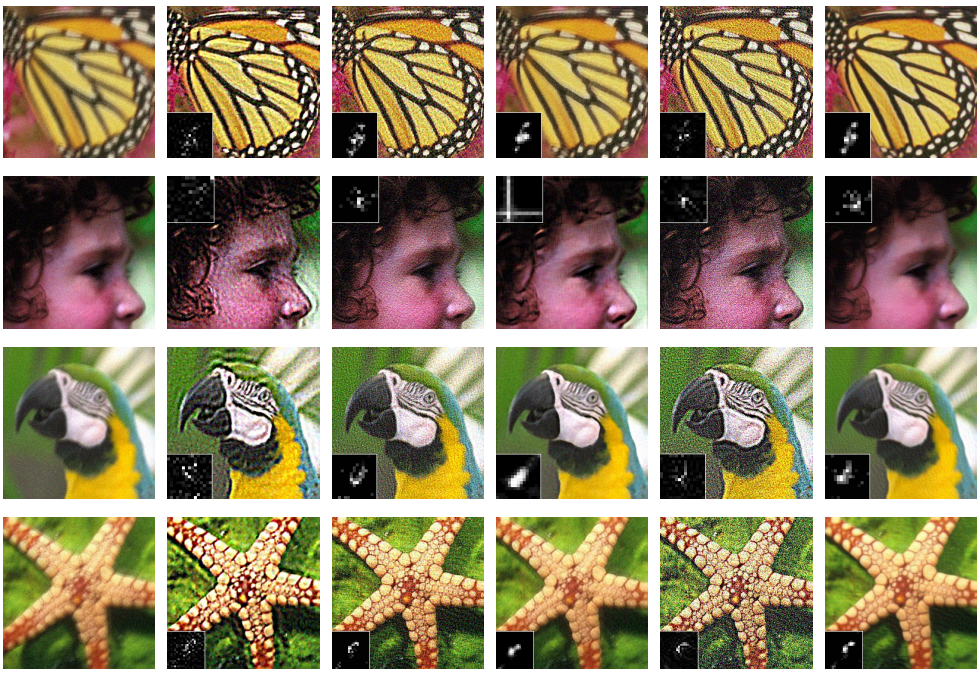
Figure 6.7 Some examples of deblurred images and estimated kernels for im.1 - im.4 in DATA SET 1. First row: image 1 using kernel 4. Second row: image 2 using kernel 8. Third row: image 3 using kernel 7. Fourth row: image 4 using kernel 2. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

(a) (b) (c) (d) (e) (f)

Figure 6.8 Some examples of deblurred images and estimated kernels for im.5 - im.8 in DATA SET 2. First row: image 5 using kernel 1. Second row: image 6 using kernel 5. Third row: image 7 using kernel 3. Fourth row: image 8 using kernel 6. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

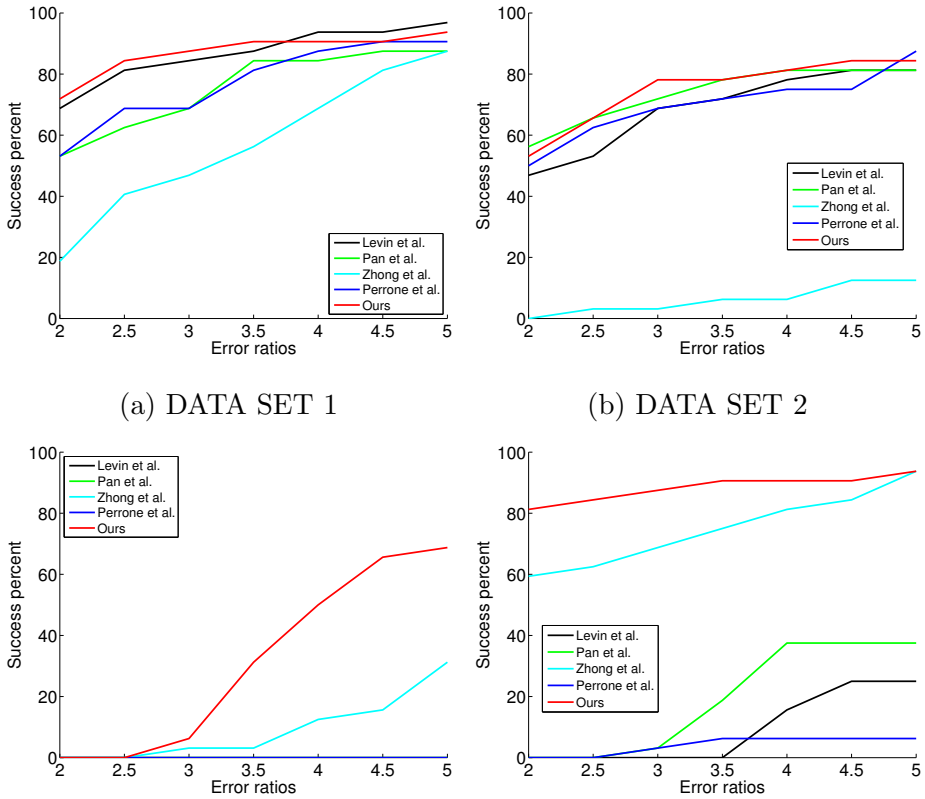
(a) (b) (c) (d) (e) (f)

Figure 6.9 Some examples of deblurred images and estimated kernels for im.1 - im.4 in DATA SET 3. First row: image 1 using kernel 4. Second row: image 2 using kernel 8. Third row: image 3 using kernel 7. Fourth row: image 4 using kernel 2. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c) (d) (e) (f)

Figure 6.10 Some examples of deblurred images and estimated kernels for im.5 - im.8 in DATA SET 4. First row: image 5 using kernel 1. Second row: image 6 using kernel 5. Third row: image 7 using kernel 3. Fourth row: image 8 using kernel 6. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) DATA SET 1

(b) DATA SET 2

(c) DATA SET 3

(d) DATA SET 4

Figure 6.11 Cumulative error ratio across test examples.

For each images in DATA SET 1, we computed SSD of the kernels and SSD, PSNR, and SSIM of the latent images. The results are listed in Tables 6.2, 6.3, 6.4, and 6.5, respectively. For each row, the highest performance was marked as a bold font. At the bottom row, the number of the highest values for each method is listed. We also computed the statistics for each of the Tables and illustrated them as box plots in Figures 6.12, 6.13, 6.14, and 6.15. In the figures, the red horizontal lines denote the median, and the red dots denote the mean of the data. The lower SSD means the higher performance, and the higher PSNR and SSIM mean the higher performance. The results of our method were compared with those of Levin *et al.* [46], Pan *et al.* [58], Zhong *et al.* [88], and Perrone *et al.* [61].

The same evaluations were performed on DATA SETs 2, 3, and 4. For each image in DATA SET 2, SSD of the kernels and SSD, PSNR, and SSIM of the latent images are listed in Tables 6.6, 6.7, 6.8, and 6.9, respectively. Their box plots are illustrated in Figures 6.16, 6.17, 6.18, and 6.19. For each image in DATA SET 3, SSD of the kernels and SSD, PSNR, and SSIM of the latent images are listed in Tables 6.10, 6.11, 6.12, and 6.13, respectively. Their box plots are illustrated in Figures 6.20, 6.21, 6.22, and 6.23. For each image in DATA SET 4, SSD of the kernels and SSD, PSNR, and SSIM of the latent images are listed in Tables 6.14, 6.15, 6.16, and 6.17, respectively. Their box plots are illustrated in Figures 6.24, 6.25, 6.26, and 6.27.

Table 6.2 SSD of the kernels for DATA SET 1.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	0.0052	0.0026	0.0058	**0.0023**	0.0028
	2	0.0052	**0.0021**	0.0203	0.0030	0.0023
	3	**0.0014**	0.0015	0.0031	0.0021	0.0018
	4	0.0088	0.0177	0.0072	**0.0037**	0.0171
	5	0.0030	0.0024	0.0077	**0.0018**	0.0026
	6	0.0160	0.0256	0.0277	**0.0142**	0.0235
	7	0.0049	0.0185	0.0326	**0.0014**	0.0154
	8	0.0045	0.0041	0.0055	**0.0015**	0.0041
2	1	**0.0021**	0.0027	0.0072	0.0031	0.0021
	2	0.0039	**0.0024**	0.0141	0.0056	0.0028
	3	0.0021	**0.0007**	0.0039	0.0031	0.0010
	4	0.0158	0.0213	0.0102	**0.0030**	0.0195
	5	**0.0015**	0.0022	0.0070	0.0337	0.0022
	6	0.0276	0.0328	0.0285	**0.0254**	0.0303
	7	0.0099	0.0347	0.0047	**0.0027**	0.0313
	8	0.0023	0.0259	0.0071	**0.0012**	0.0236
3	1	0.0031	0.0048	0.0065	**0.0022**	0.0050
	2	**0.0022**	0.0030	0.0176	0.0025	0.0029
	3	**0.0012**	0.0018	0.0038	0.0018	0.0019
	4	0.0059	0.0174	0.0046	**0.0034**	0.0168
	5	**0.0015**	0.0038	0.0066	0.0032	0.0039
	6	**0.0016**	0.0214	0.0248	0.0034	0.0214
	7	0.0336	0.0279	0.0033	**0.0026**	0.0261
	8	0.0231	0.0045	0.0028	**0.0023**	0.0054
4	1	0.0072	0.0056	0.0103	**0.0051**	0.0053
	2	0.0057	0.0039	0.0225	0.0053	**0.0037**
	3	0.0089	**0.0029**	0.0040	0.0037	0.0034
	4	**0.0049**	0.0311	0.0172	0.0058	0.0247
	5	0.0073	0.0055	0.0063	**0.0042**	0.0065
	6	0.0273	0.0286	0.0311	0.0299	**0.0257**
	7	0.0265	0.0195	0.0092	**0.0083**	0.0182
	8	0.0147	0.0131	0.0126	**0.0047**	0.0140
#		8	4	0	18	2

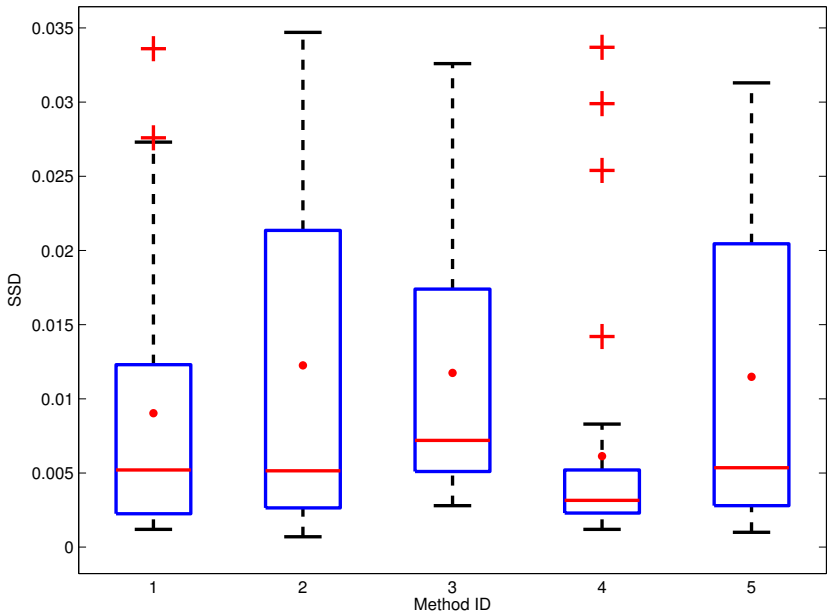


Figure 6.12 Box plot of SSD of the kernels for DATA SET 1. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the kernel estimation measured by SSD for DATA SET 1, method of Perrone *et al.* [61] attained the highest performance, as shown in Table 6.2 and in Figure 6.12. In the number of the highest performance for each image, Perrone *et al.* showed the highest frequency, which means the highest performance. Levin *et al.* was the second. Pan *et al.* was the third, and ours was the fourth. Zhong *et al.* showed the lowest frequency. In the median, Perrone *et al.* showed the smallest value, which means the highest performance. Levin *et al.* and Pan *et al.*, and ours were similar. Zhong *et al.* showed a little larger value than others. In the mean, Perrone *et al.* also showed the smallest value, which means the highest performance. Levin *et al.* was the second. Zhong *et al.* and ours showed the similar values. About the mean, the result of Pan *et al.* was the largest, which means the lowest performance.

Table 6.3 SSD of the latent images for DATA SET 1.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	52.2490	**33.3064**	91.4253	51.3802	33.9063
	2	51.0729	36.5374	183.5208	47.8174	**35.4151**
	3	31.6459	**25.6409**	46.6366	30.1610	26.8503
	4	144.2454	163.7985	267.8554	**122.6167**	150.4657
	5	35.0979	32.6674	51.0296	**28.2305**	31.7801
	6	**36.8819**	52.0789	93.6770	76.6535	37.5382
	7	**70.4127**	92.4786	133.8277	94.1283	71.0721
	8	65.4778	76.8485	180.0580	103.9806	**65.3978**
2	1	54.6202	42.7638	132.2725	77.5388	**40.2370**
	2	74.4258	**37.3162**	192.8802	128.5723	38.6733
	3	47.3133	**21.3118**	65.9254	51.6026	22.7344
	4	111.9309	245.5966	242.8374	**106.6205**	226.9676
	5	**25.2378**	31.1457	61.4161	82.0292	27.0515
	6	**32.7428**	71.6787	102.4899	46.6707	54.7620
	7	67.3492	97.8291	113.2308	90.7752	**56.0526**
	8	**63.1845**	149.5712	176.8159	94.9602	68.8850
3	1	40.8649	31.3985	94.2357	60.5327	**27.8148**
	2	51.1409	34.6341	176.6163	53.6547	**32.5601**
	3	**17.9183**	20.0558	46.3894	30.8590	19.1972
	4	78.2859	52.2474	215.2067	69.8633	**48.6201**
	5	**18.5809**	22.8650	37.1307	34.3432	19.5567
	6	**22.5086**	41.8723	91.1916	44.5020	32.5735
	7	**35.3393**	129.2660	66.3625	61.9436	50.0694
	8	**42.1534**	117.2363	100.7481	76.7820	51.8307
4	1	83.5337	35.0584	107.2201	75.8149	**33.4784**
	2	137.6859	97.1924	250.0331	162.0755	**73.6380**
	3	69.9277	**20.8188**	30.1710	42.8016	24.1757
	4	**106.8158**	445.6475	415.9395	196.7859	457.9862
	5	56.4325	23.7097	**21.4217**	37.5110	28.3795
	6	79.3189	65.7172	93.0809	109.4044	**54.3188**
	7	130.3128	159.2743	199.7980	155.0895	**97.2045**
	8	126.0181	183.9150	247.7136	155.5907	**111.8702**
#		11	5	1	3	12

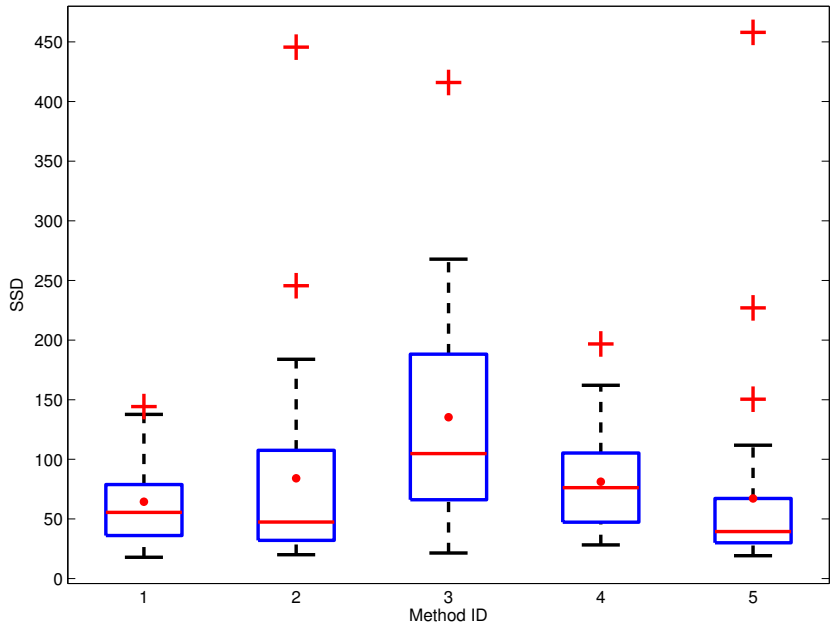


Figure 6.13 Box plot of SSD of the latent images for DATA SET 1. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSD for DATA SET 1, ours attained the highest performance, as shown in Table 6.3 and in Figure 6.13. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Levin *et al.* showed the second highest frequency. Pan *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* showed the lowest frequency. In the median, ours showed the smallest value, which means the highest performance. Pan *et al.* was the second Levin *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* showed the largest value. In the mean, Levin *et al.* showed the smallest value, which means the highest performance. Our was the second. Perrone *et al.* was the third, and Pan *et al.* was the fourth. Zhong *et al.* was the largest, which means the lowest performance.

Table 6.4 PSNR of the latent images for DATA SET 1.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	30.60	**32.53**	28.26	30.66	32.46
	2	30.70	32.14	25.38	30.97	**32.28**
	3	32.78	**33.66**	31.13	32.97	33.47
	4	26.26	25.79	23.87	**26.95**	26.18
	5	32.32	32.63	30.79	**33.25**	32.74
	6	**32.11**	30.61	28.08	28.95	32.03
	7	**29.34**	28.21	26.59	28.07	29.30
	8	29.64	28.98	25.64	27.63	**29.65**
2	1	30.40	31.46	26.59	28.88	**31.72**
	2	29.09	**32.05**	24.94	26.79	31.90
	3	31.02	**34.47**	29.60	30.65	34.19
	4	27.32	23.98	23.99	**27.60**	24.37
	5	**33.74**	32.83	29.90	28.64	33.44
	6	**32.61**	29.23	27.69	31.07	30.40
	7	29.50	27.88	27.27	28.25	**30.34**
	8	**29.78**	26.06	25.42	28.03	29.41
3	1	31.65	32.80	28.06	29.96	**33.32**
	2	30.70	32.39	25.32	30.48	**32.65**
	3	**35.22**	34.76	31.26	32.87	34.94
	4	28.87	30.74	24.54	29.45	**31.03**
	5	**35.06**	34.17	32.13	32.41	34.85
	6	**34.24**	31.59	28.19	31.30	32.66
	7	**32.31**	26.76	29.57	29.91	30.81
	8	**31.55**	27.24	27.79	28.97	30.70
4	1	28.66	32.32	27.66	29.02	**32.54**
	2	26.44	27.92	23.96	25.70	**29.12**
	3	29.63	**34.69**	32.99	31.45	34.09
	4	**27.50**	21.29	21.65	24.98	21.18
	5	30.58	34.08	**34.45**	32.02	33.40
	6	28.93	29.61	28.12	27.41	**30.48**
	7	26.68	25.77	24.99	25.89	**27.95**
	8	26.81	25.15	23.96	25.88	**27.31**
#		11	5	1	3	12

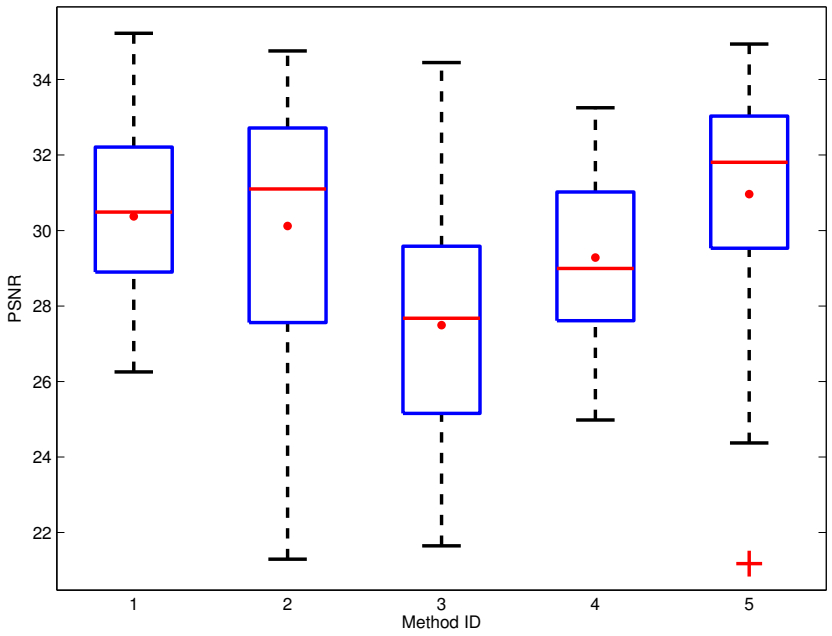


Figure 6.14 Box plot of PSNR of the latent images for DATA SET 1. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by PSNR for DATA SET 1, ours attained the highest performance, as shown in Table 6.4 and in Figure 6.14. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Levin *et al.* showed the second highest frequency. Pan *et al.* was the third, and Perrone *et al.* showed the fourth. Zhong *et al.* showed the lowest frequency. In the median, ours showed the largest value, which means the highest performance. Levin *et al.* was the second. Pan *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* showed the smallest value. In the mean, ours also showed the largest value, which means the highest performance. Levin *et al.* was the second. Pan *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* was the smallest, which means the lowest performance.

Table 6.5 SSIM of the latent images for DATA SET 1.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	0.9100	**0.9419**	0.8500	0.9145	0.9407
	2	0.9157	**0.9412**	0.7708	0.9246	0.9411
	3	0.9396	**0.9560**	0.9032	0.9433	0.9497
	4	0.8515	0.8489	0.7423	**0.8700**	0.8522
	5	0.9531	0.9614	0.9105	0.9570	**0.9615**
	6	**0.9436**	0.9254	0.8658	0.9119	0.9363
	7	0.9046	0.8938	0.8574	0.8975	**0.9065**
	8	0.9045	0.9035	0.7923	0.8815	**0.9123**
2	1	0.8959	0.9238	0.7947	0.8831	**0.9275**
	2	0.8804	**0.9321**	0.7504	0.8354	0.9314
	3	0.9073	**0.9548**	0.8867	0.9165	0.9503
	4	0.8319	0.7541	0.6986	**0.8685**	0.7644
	5	**0.9535**	0.9526	0.9051	0.8932	0.9527
	6	**0.9438**	0.9134	0.8319	0.9344	0.9296
	7	0.9033	0.8892	0.8421	0.8883	**0.9248**
	8	**0.8993**	0.8355	0.7570	0.8883	0.8945
3	1	0.9386	0.9513	0.8686	0.9122	**0.9559**
	2	0.9340	**0.9527**	0.7905	0.9307	0.9503
	3	0.9673	0.9682	0.9199	0.9539	**0.9694**
	4	0.9008	0.9243	0.7570	0.9151	**0.9315**
	5	**0.9743**	0.9699	0.9393	0.9515	0.9726
	6	**0.9613**	0.9384	0.8653	0.9377	0.9517
	7	**0.9490**	0.8582	0.9092	0.9280	0.9243
	8	**0.9377**	0.8535	0.8465	0.9163	0.9252
4	1	0.8822	**0.9363**	0.8464	0.8789	0.9345
	2	0.8387	0.8751	0.7239	0.8007	**0.9009**
	3	0.9119	**0.9613**	0.9435	0.9245	0.9574
	4	**0.8640**	0.6342	0.6159	0.7735	0.6228
	5	0.9294	**0.9636**	0.9635	0.9417	0.9576
	6	0.9054	0.9115	0.8669	0.8544	**0.9236**
	7	0.8734	0.8595	0.8027	0.8490	**0.8924**
	8	**0.8667**	0.8112	0.7556	0.8115	0.8660
#		10	9	0	2	11

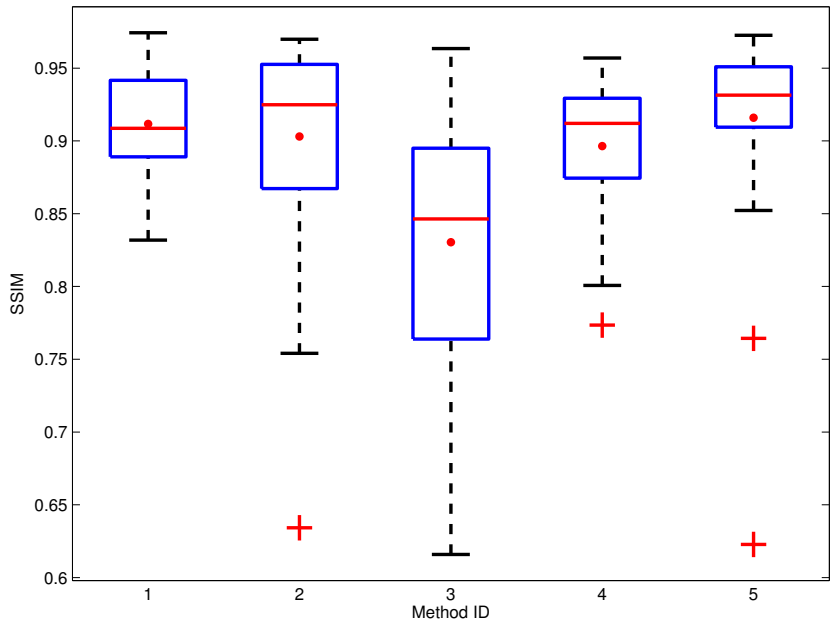


Figure 6.15 Box plot of SSIM of the latent images for DATA SET 1. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61],

In the accuracy on the image estimation measured by SSIM for DATA SET 1, ours attained the highest performance, as shown in Table 6.5 and in Figure 6.15. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Levin *et al.* showed the second highest frequency. Pan *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* showed the lowest frequency. In the median, ours showed the largest value, which means the highest performance. Pan *et al.* was the second Perrone *et al.* was the third, and Levin *et al.* was the fourth. Zhong *et al.* showed the smallest value. In the mean, ours also showed the largest value, which means the highest performance. Levin *et al.* showed the second smallest value. Pan *et al.* was the third, and Perrone *et al.* was the fourth. Zhong *et al.* was the smallest, which means the lowest performance.

Table 6.6 SSD of the kernels for DATA SET 2.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	0.0015	0.0046	0.0059	**0.0013**	0.0037
	2	0.0022	**0.0010**	0.0091	0.0019	0.0010
	3	0.0012	**0.0006**	0.0045	0.0010	0.0007
	4	0.0195	0.0250	0.0139	**0.0035**	0.0186
	5	0.0011	**0.0005**	0.0073	0.0017	0.0005
	6	0.0380	0.0295	0.0489	**0.0278**	0.0297
	7	**0.0022**	0.0348	0.0148	0.0211	0.0344
	8	0.0329	0.0302	0.0219	**0.0054**	0.0292
6	1	0.0096	**0.0037**	0.0269	0.0101	0.0046
	2	**0.0037**	0.0071	0.0323	0.0046	0.0080
	3	0.0031	0.0022	0.0217	0.0030	**0.0018**
	4	0.0049	0.0160	0.0153	**0.0028**	0.0120
	5	0.0037	0.0030	0.0415	0.0242	**0.0029**
	6	**0.0066**	0.0426	0.0318	0.0123	0.0363
	7	**0.0031**	0.0099	0.0239	0.0047	0.0047
	8	0.0049	0.0017	0.0190	0.0020	**0.0014**
7	1	0.0074	**0.0020**	0.0103	0.0024	0.0022
	2	0.0040	0.0038	0.0161	**0.0020**	0.0039
	3	0.0080	0.0016	0.0050	0.0024	**0.0014**
	4	0.0128	0.0186	0.0110	**0.0015**	0.0179
	5	0.0031	0.0026	0.0070	0.0129	**0.0024**
	6	0.0113	0.0175	0.0438	**0.0043**	0.0179
	7	0.0074	0.0293	0.0051	**0.0018**	0.0283
	8	0.0050	0.0020	0.0059	0.0036	**0.0017**
8	1	0.0018	0.0026	0.0076	**0.0016**	0.0023
	2	0.0012	0.0015	0.0217	**0.0008**	0.0015
	3	**0.0008**	0.0014	0.0056	0.0024	0.0009
	4	**0.0011**	0.0208	0.0050	0.0013	0.0195
	5	**0.0015**	0.0018	0.0053	0.0060	0.0015
	6	**0.0006**	0.0133	0.0369	0.0049	0.0108
	7	**0.0013**	0.0267	0.0353	0.0016	0.0266
	8	0.0008	**0.0007**	0.0035	0.0016	0.0008
#		9	6	0	11	6

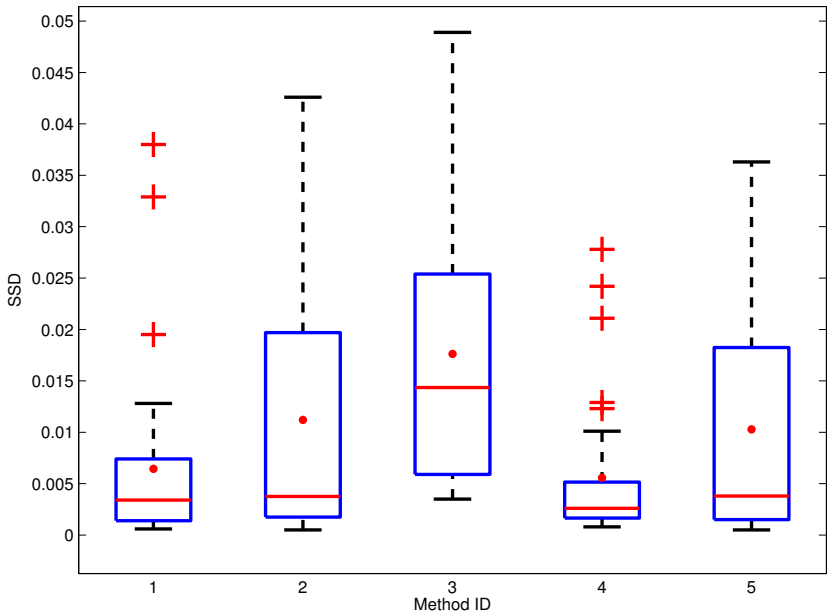


Figure 6.16 Box plot of SSD of the kernels for DATA SET 2. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the kernel estimation measured by SSD for DATA SET 2, method of Perrone *et al.* [61] attained the highest performance, as shown in Table 6.6 and in Figure 6.16. In the number of the highest performance for each image , Perrone *et al.* showed the highest frequency, which means the highest performance. Levin *et al.* was the second. Pan *et al.* and ours showed the same frequency. Zhong *et al.* showed the lowest frequency. In the median, Perrone *et al.* showed the smallest value, which means the highest performance. Levin *et al.* and Pan *et al.*, and ours are similar. Zhong *et al.* showed the largest value. In the mean, Perrone *et al.* and Levin *et al.* showed the similarly small value, which means the highest performance. Pan *et al.* and ours showed the similar values. Zhong *et al.* showed the largest, which means the lowest performance.

Table 6.7 SSD of the latent images for DATA SET 2.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	**103.2979**	131.4954	608.2955	107.1912	108.3109
	2	146.2413	**68.1638**	959.1504	462.3361	70.2111
	3	62.8682	**46.6748**	280.4911	55.9188	49.5830
	4	227.3894	2562.1099	856.6930	**146.4961**	2037.6583
	5	39.4993	46.5844	244.5260	**28.7576**	47.4331
	6	**88.4895**	542.4260	922.2112	610.8568	374.5473
	7	**278.4299**	1366.7452	1007.6686	330.3621	1484.5543
	8	3196.9938	2303.1426	997.4367	**291.5479**	2290.8709
6	1	43.5472	22.4548	285.6263	71.0073	**21.9971**
	2	26.5399	25.6025	248.2248	34.7913	**24.1417**
	3	22.4552	14.9536	232.6613	18.3482	**12.8323**
	4	30.0262	67.6913	669.1914	**20.7389**	54.9210
	5	19.8985	**10.2000**	172.0336	13.2905	10.6086
	6	24.6038	62.0930	631.5628	**9.9761**	27.5038
	7	24.7915	36.7071	539.8547	25.0469	**15.7723**
	8	33.5556	16.4978	475.2183	16.0590	**13.0251**
7	1	129.1273	**41.5341**	538.5844	97.5552	56.8937
	2	99.7116	**48.2413**	491.5401	64.9813	58.0327
	3	92.8109	**30.6276**	221.7883	53.4379	36.2378
	4	143.0129	106.5130	560.3680	**84.1775**	100.6548
	5	48.7407	28.8012	236.1181	155.5615	**27.6039**
	6	202.8644	186.8844	300.2186	157.8003	**139.9493**
	7	137.8796	67.2215	420.1222	**52.5164**	110.3650
	8	153.2238	53.0271	536.4149	126.5167	**45.0229**
8	1	56.5416	46.7830	265.1232	59.5119	**41.4850**
	2	58.7523	**41.2177**	297.1743	49.9579	41.4256
	3	42.4010	31.3452	136.4686	39.5551	**27.2107**
	4	87.3018	128.2786	301.7192	**85.9102**	129.5547
	5	29.2163	24.2027	81.3644	35.5548	**21.5335**
	6	**26.7184**	60.6264	126.8707	42.6332	81.6804
	7	**65.6500**	83.1618	100.1892	89.2928	121.4648
	8	**63.9988**	68.6550	265.4249	70.1103	79.9406
#		6	7	0	8	11

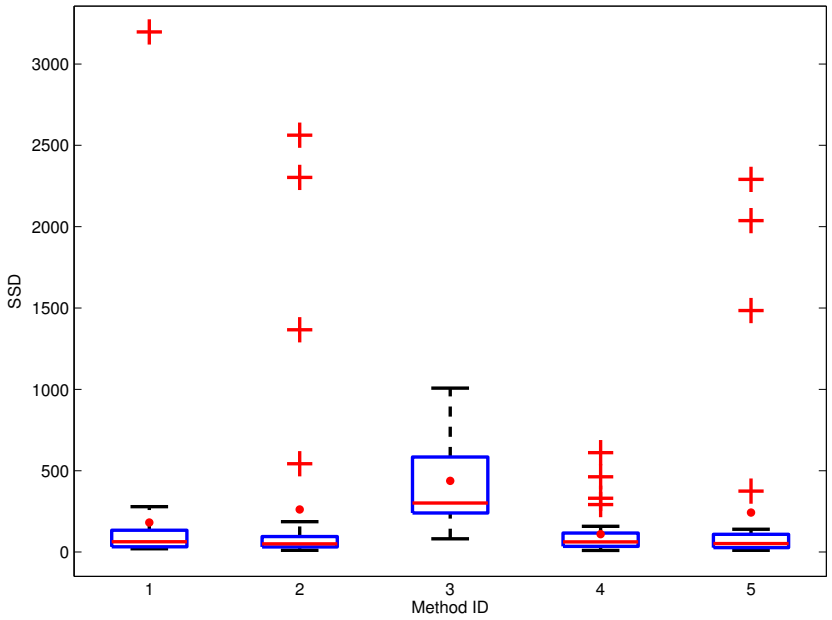


Figure 6.17 Box plot of SSD of the latent images for DATA SET 2. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSD for DATA SET 2, ours attained the highest performance, as shown in Table 6.7 and in Figure 6.17. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Perrone *et al.* showed the second highest frequency. Pan *et al.* was the third, and Levin *et al.* was the fourth. Zhong *et al.* showed the lowest frequency. In the median, Levin *et al.*, Pan *et al.*, Perrone *et al.*, and ours were similar. Zhong *et al.* showed the largest value. In the mean, Levin *et al.* and Perrone *et al.* showed the similarly small value, which means the highest performance. Pan *et al.* and ours were also similar. Zhong *et al.* was the largest, which means the lowest performance.

Table 6.8 PSNR of the latent images for DATA SET 2.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	**27.58**	26.64	20.14	27.49	27.44
	2	26.15	**29.35**	18.19	21.38	29.31
	3	29.68	**31.08**	23.43	30.24	30.78
	4	24.39	13.98	18.73	**26.16**	14.97
	5	31.79	31.19	24.05	**33.02**	31.11
	6	**28.43**	20.64	18.32	20.09	22.26
	7	**23.62**	16.69	18.07	22.76	16.32
	8	13.04	14.43	18.11	**23.39**	14.45
6	1	30.64	33.37	22.78	28.41	**33.37**
	2	32.67	32.95	23.43	31.31	**33.14**
	3	33.48	34.71	23.69	33.94	**35.49**
	4	32.23	28.78	19.07	**33.37**	29.74
	5	33.96	**36.65**	24.98	35.61	36.47
	6	32.78	28.90	19.41	**36.37**	32.36
	7	33.00	31.46	20.06	32.97	**34.92**
	8	31.74	34.59	20.58	34.55	**35.64**
7	1	26.48	**31.37**	20.44	27.80	29.98
	2	27.59	**30.76**	20.86	29.50	29.94
	3	27.86	**32.76**	24.28	30.34	32.03
	4	25.94	27.24	20.06	**28.15**	27.49
	5	30.74	32.99	24.09	25.79	**33.19**
	6	24.53	24.95	22.93	25.69	**26.16**
	7	26.24	29.28	21.33	**30.33**	27.23
	8	25.75	30.20	20.26	26.68	**30.97**
8	1	30.32	31.09	23.93	30.07	**31.57**
	2	30.15	**31.72**	23.34	30.78	31.63
	3	31.49	32.75	26.68	31.77	**33.35**
	4	**28.51**	26.83	23.39	28.50	26.79
	5	33.22	33.95	28.88	32.27	**34.41**
	6	**33.50**	30.15	26.98	31.42	28.93
	7	**29.76**	28.87	27.98	28.42	27.26
	8	**29.89**	29.70	24.01	29.49	29.10
#		7	7	0	7	11

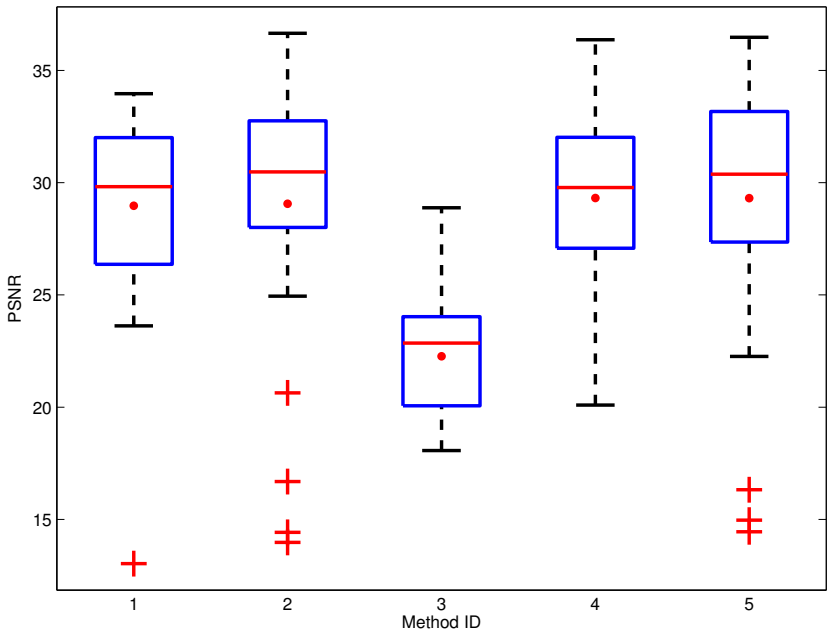


Figure 6.18 Box plot of PSNR of the latent images for DATA SET 2. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by PSNR for DATA SET 2, ours attained the highest performance, as shown in Table 6.8 and in Figure 6.18. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Levin *et al.*, Pan *et al.*, and Perrone *et al.* showed the same frequency. Zhong *et al.* showed the lowest frequency. In the median, Pan *et al.* and ours showed the similar value, attaining the first and the second largest value. Levin *et al.* and Perrone *et al.* also showed the similar value, attaining the third and the fourth largest value. Zhong *et al.* showed the smallest value. In the mean, Perrone *et al.* and ours also showed the similar value, attaining the first and the second largest value. Pan *et al.* and Levin *et al.* also showed the similar value, attaining the third and the fourth largest value. Zhong *et al.* was the smallest.

Table 6.9 SSIM of the latent images for DATA SET 2.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	0.9020	0.8839	0.6685	**0.9324**	0.8964
	2	0.8783	0.9338	0.5602	0.7264	**0.9451**
	3	0.9349	0.9598	0.8037	**0.9626**	0.9538
	4	0.8700	0.2659	0.6194	**0.9081**	0.3819
	5	0.9664	0.9714	0.8590	**0.9715**	0.9706
	6	**0.9361**	0.7240	0.5891	0.7653	0.7751
	7	0.8487	0.5159	0.6890	**0.8537**	0.4743
	8	0.1758	0.2634	0.6599	**0.8300**	0.2704
6	1	0.8117	**0.8792**	0.5922	0.7803	0.8690
	2	0.8522	**0.8769**	0.6287	0.8564	0.8733
	3	0.8836	0.8824	0.6773	**0.9034**	0.8969
	4	0.8435	0.7877	0.4068	**0.8957**	0.8349
	5	0.9131	**0.9411**	0.6780	0.9355	0.9342
	6	0.8610	0.7462	0.4152	**0.9426**	0.8629
	7	0.8898	0.8885	0.4466	0.9123	**0.9163**
	8	0.8513	0.9001	0.4717	**0.9183**	0.9125
7	1	0.8831	**0.9462**	0.7343	0.9261	0.9328
	2	0.9014	0.9336	0.7351	**0.9406**	0.9244
	3	0.9078	**0.9520**	0.8676	0.9459	0.9475
	4	0.8669	0.8888	0.7000	**0.9198**	0.8903
	5	0.9490	**0.9644**	0.8729	0.8911	0.9642
	6	0.8782	0.9002	0.8320	**0.9068**	0.9063
	7	0.8982	0.9261	0.7550	**0.9487**	0.9101
	8	0.8753	0.9322	0.7056	0.8932	**0.9391**
8	1	0.9004	0.9210	0.8109	**0.9272**	0.9203
	2	0.8940	0.9357	0.7458	**0.9403**	0.9246
	3	0.9162	0.9365	0.8551	0.9389	**0.9404**
	4	0.8725	0.8472	0.7843	**0.9119**	0.8419
	5	0.9556	**0.9596**	0.9133	0.9523	0.9594
	6	0.9505	0.9390	0.8753	**0.9527**	0.9236
	7	0.9126	0.9274	0.8995	**0.9283**	0.9133
	8	0.9082	0.9264	0.8388	**0.9281**	0.9270
#		1	7	0	20	4

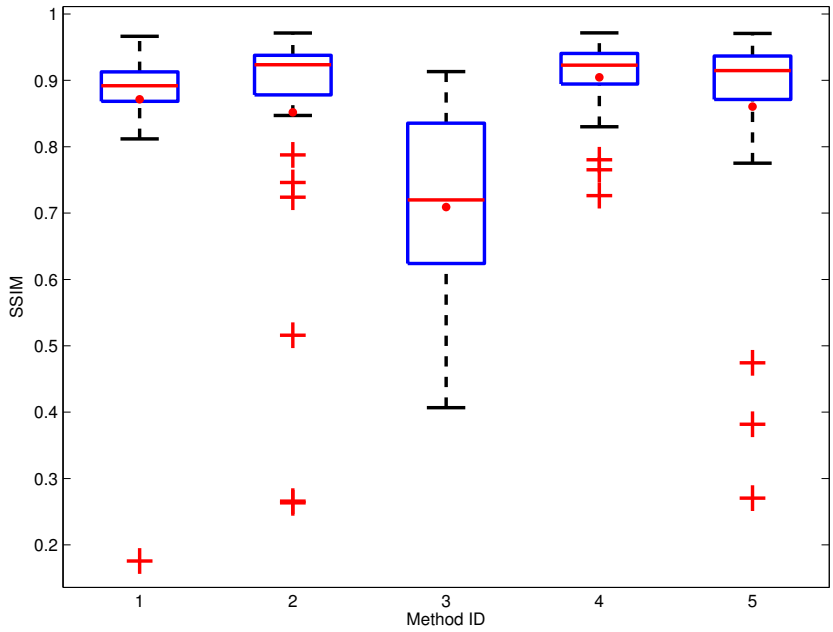


Figure 6.19 Box plot of SSIM of the latent images for DATA SET 2. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSIM for DATA SET 2, Perrone *et al.* attained the highest performance, as shown in Table 6.9 and in Figure 6.19. In the number of the highest performance for each image, Perrone *et al.* showed the highest frequency, which means the highest performance. Pan *et al.* showed the second highest frequency. Ours was the third, and Levin *et al.* was the fourth. Zhong *et al.* showed the lowest frequency. In the median, Perrone *et al.*, Pan *et al.*, and ours showed the similar value, attaining the first, the second, and the third largest value. Levin *et al.* was the fourth. Zhong *et al.* showed the smallest value. In the mean, Perrone *et al.* showed the largest value, which means the highest performance. Pan *et al.* was the second. Levin *et al.* and ours was similar. Zhong *et al.* was the smallest, which means the lowest performance.

Table 6.10 SSD of the kernels for DATA SET 3.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	0.0222	0.0252	**0.0058**	0.0166	0.0098
	2	0.0183	0.0219	0.0212	0.0123	**0.0056**
	3	0.0193	0.0267	**0.0029**	0.0165	0.0065
	4	0.0187	0.0257	**0.0055**	0.0105	0.0223
	5	0.0259	0.0173	0.0083	0.0194	**0.0056**
	6	0.0257	0.0337	0.0294	**0.0246**	0.0272
	7	0.0237	0.0355	0.0283	**0.0188**	0.0329
	8	0.0143	0.0224	**0.0044**	0.0105	0.0221
2	1	0.0173	0.0205	**0.0090**	0.0184	0.0109
	2	0.0237	0.0259	0.0183	0.0204	**0.0148**
	3	0.0288	0.0295	**0.0031**	0.0312	0.0119
	4	0.0189	0.0356	**0.0074**	0.0097	0.0278
	5	0.0254	0.0208	**0.0089**	0.1222	0.0099
	6	0.0324	0.0318	**0.0316**	0.0359	0.0674
	7	0.0241	0.0385	0.0144	**0.0121**	0.0469
	8	0.0148	0.0348	**0.0083**	0.0097	0.0312
3	1	0.0182	0.0234	**0.0086**	0.0180	0.0114
	2	0.0194	0.0224	0.0170	0.0153	**0.0120**
	3	0.0217	0.0220	**0.0035**	0.0255	0.0090
	4	0.0176	0.0241	**0.0049**	0.0087	0.0202
	5	0.0441	0.0208	0.0074	0.0158	**0.0067**
	6	0.0261	0.0270	0.0268	**0.0193**	0.0378
	7	0.0232	0.0338	**0.0042**	0.0111	0.0303
	8	0.0237	0.0214	**0.0039**	0.0112	0.0140
4	1	0.0239	0.0400	**0.0127**	0.0233	0.0224
	2	0.0203	0.0346	0.0220	**0.0185**	0.0217
	3	0.0262	0.0269	**0.0050**	0.0212	0.0169
	4	0.0181	0.0303	0.0156	**0.0092**	0.0372
	5	0.0375	0.0268	**0.0064**	0.0198	0.0127
	6	0.0304	0.0370	**0.0271**	0.0317	0.0518
	7	0.0239	0.0460	**0.0099**	0.0226	0.0571
	8	0.0187	0.0354	**0.0123**	0.0136	0.0504
#		0	0	21	6	5

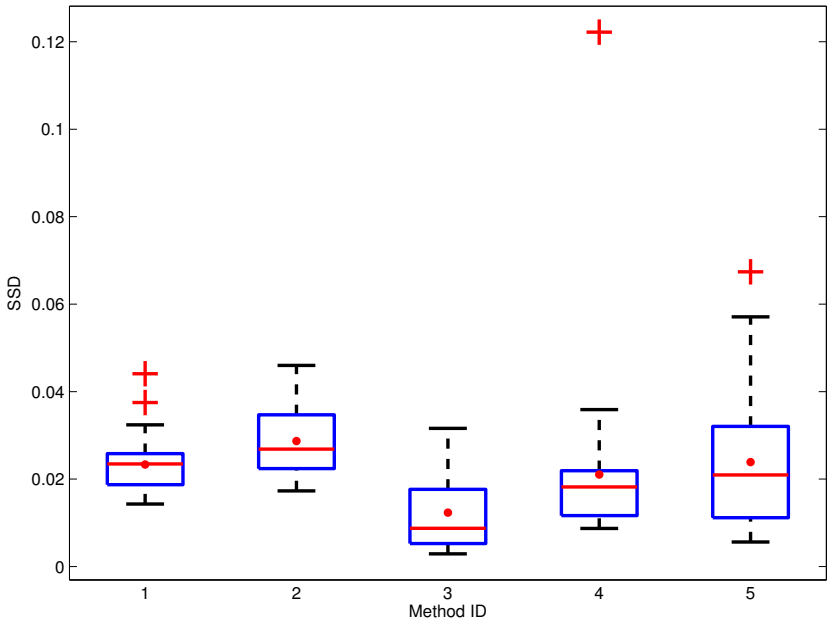


Figure 6.20 Box plot of SSD of the kernels for DATA SET 3. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the kernel estimation measured by SSD for DATA SET 3, method of Zhong *et al.* [88] attained the highest performance, as shown in Table 6.10 and in Figure 6.20. In the number of the highest performance for each image , Zhong *et al.* showed the highest frequency, which means the highest performance. Perrone *et al.* was the second. Ours was the third. Levin *et al.* and Zhong *et al.* showed the lowest frequency. In the median, Zhong *et al.* showed the smallest value, which means the highest performance. Perrone *et al.* was the second, Ours was the third. Levin *et al.* was the fourth. Pan *et al.* showed the largest value. In the mean, Zhong *et al.* showed the smallest value, which means the highest performance. Perrone *et al.* was the second. Levin *et al.* was the third. Ours was the fourth. Pan *et al.* showed the largest, which means the lowest performance.

Table 6.11 SSD of the latent images for DATA SET 3.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	2587.5373	2118.9132	168.1978	5156.7590	**115.4084**
	2	2112.3995	1824.1946	298.0046	8581.7862	**123.6511**
	3	1630.6037	1551.1461	112.6953	4913.1435	**96.9347**
	4	4763.8275	3255.4210	320.4285	11424.8922	**277.5148**
	5	1031.9317	1369.1739	135.2917	4084.0903	**77.6950**
	6	2081.2357	2463.1782	174.0186	7653.7883	**75.2791**
	7	6737.6556	2220.3752	221.4995	4212.4898	**112.5509**
	8	3327.7010	3085.9272	266.7816	10558.4112	**136.0465**
2	1	2953.6289	1643.1046	221.8304	4996.3770	**190.0786**
	2	1364.3292	1658.1019	341.7245	5583.5794	**208.8524**
	3	1846.3583	1405.7844	**134.2008**	2510.4402	161.4137
	4	2356.0732	2480.2501	325.9115	12335.3742	**292.7523**
	5	1571.2484	1088.4469	**149.2452**	562.6099	158.5108
	6	1927.2209	2150.0046	186.4937	3476.1175	**173.3697**
	7	2396.8700	2138.2097	212.1253	6808.8438	**161.9155**
	8	2596.3527	2138.7784	233.6417	11250.0540	**201.4915**
3	1	3114.6535	1829.4631	181.0874	5904.1443	**146.8629**
	2	1735.5513	1820.7872	298.9414	7597.0942	**172.3665**
	3	1739.2193	1540.9586	110.7969	2825.5465	**98.6608**
	4	3391.0762	2911.1868	299.0703	9939.3903	**297.8128**
	5	706.9567	1305.8535	127.9217	5023.6382	**103.8559**
	6	2486.1561	2396.2274	166.0711	12086.8978	**89.5357**
	7	6035.5045	2342.2665	168.9845	5654.1223	**135.0278**
	8	1827.9294	2813.2119	201.7579	8569.2933	**182.3848**
4	1	2179.8011	1521.4828	199.9624	4884.2110	**130.6630**
	2	1918.8249	1747.0022	339.7143	7417.7301	**202.1352**
	3	1540.2770	1507.0310	98.5289	5278.2549	**61.4869**
	4	3096.2705	3881.2977	**427.3684**	8256.4437	539.1842
	5	935.6691	1223.0008	113.5280	5057.0711	**76.6010**
	6	2169.6113	2182.3682	172.1240	6092.9521	**132.0496**
	7	3289.9670	2323.1419	**251.9058**	5549.8972	281.5275
	8	1626.2145	2154.2316	**328.4587**	6944.8196	339.4928
#		0	0	5	0	27

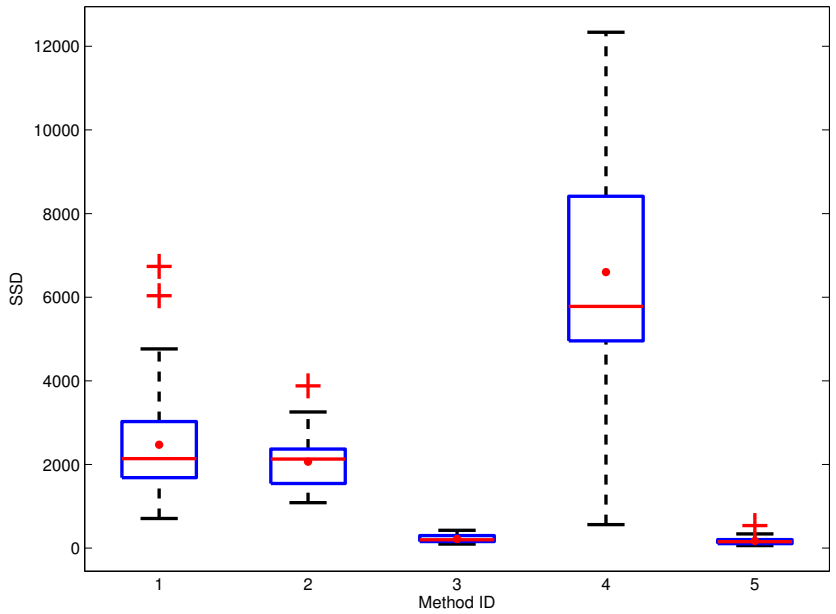


Figure 6.21 Box plot of SSD of the latent images for DATA SET 3. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSD for DATA SET 3, ours attained the highest performance, as shown in Table 6.11 and in Figure 6.21. In the number of the highest performance for each image, for almost all of the images, ours showed the highest performance. Zhong *et al.* showed the second highest frequency. Levin *et al.*, Pan *et al.*, and Perrone *et al.* showed the zero of the frequency. In the median, Zhong *et al.* and ours showed the similarly lowest value, which means the high performance. Levin *et al.* and Pan *et al.* showed the similar value, attaining the third and the fourth. Perrone *et al.* showed the largest value, which means the lowest performance. The performance was similar in the mean.

Table 6.12 PSNR of the latent images for DATA SET 3.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	16.10	16.07	25.65	12.90	**27.15**
	2	16.72	16.57	23.30	11.30	**26.86**
	3	17.79	17.09	27.38	13.03	**27.91**
	4	14.07	15.01	23.14	10.75	**23.60**
	5	19.06	17.54	26.65	13.71	**28.87**
	6	17.14	15.88	25.46	11.95	**29.01**
	7	13.76	16.18	24.43	13.31	**27.32**
	8	15.67	15.11	23.87	10.88	**26.45**
2	1	15.53	17.19	24.44	13.10	**24.99**
	2	17.97	17.09	22.61	12.65	**24.58**
	3	17.24	17.66	**26.61**	15.56	25.70
	4	16.02	15.73	22.82	10.54	**23.25**
	5	17.82	18.55	**26.16**	20.85	25.78
	6	17.30	16.44	25.20	14.28	**25.39**
	7	16.61	16.38	24.63	12.21	**25.69**
	8	16.23	16.20	24.25	10.78	**24.76**
3	1	15.33	16.36	25.29	12.30	**26.11**
	2	17.10	16.41	23.12	11.51	**25.42**
	3	17.38	16.98	27.52	14.67	**27.83**
	4	14.84	14.96	23.20	10.96	**23.27**
	5	20.34	17.57	26.86	12.91	**27.61**
	6	16.25	15.70	25.65	10.55	**28.26**
	7	13.67	15.85	25.62	12.27	**26.48**
	8	16.96	15.08	24.87	11.18	**25.24**
4	1	16.22	16.81	24.94	12.70	**26.62**
	2	16.65	16.20	22.69	11.46	**24.73**
	3	17.48	17.00	27.91	12.58	**29.89**
	4	15.10	13.35	**21.60**	11.37	20.48
	5	19.15	17.75	27.32	12.69	**28.93**
	6	16.37	15.74	25.49	12.04	**26.58**
	7	15.01	15.42	**23.96**	12.28	23.29
	8	17.05	15.59	**22.78**	11.62	22.48
#		0	0	5	0	27

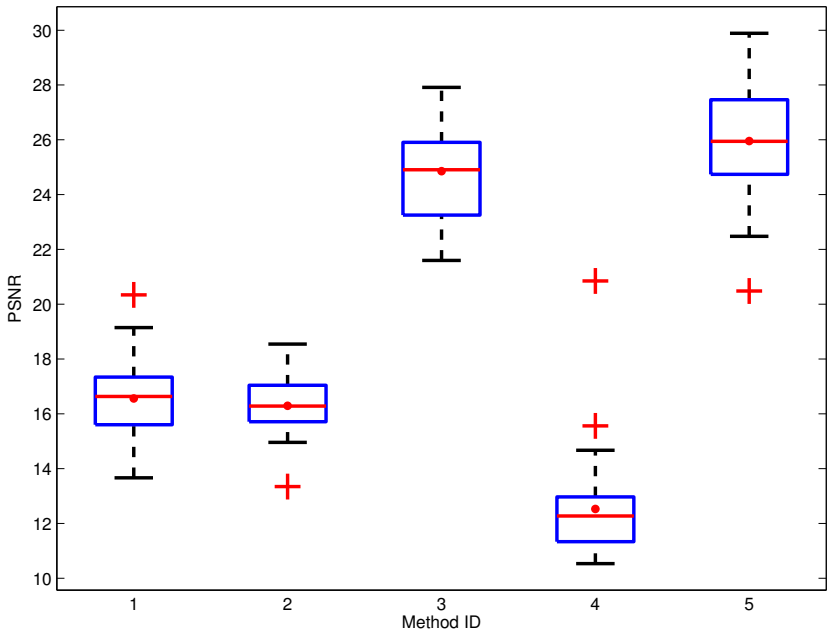


Figure 6.22 Box plot of PSNR of the latent images for DATA SET 3. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by PSNR for DATA SET 3, ours attained the highest performance, as shown in Table 6.12 and in Figure 6.22. In the number of the highest performance for each image, ours showed the highest performance for almost all of the images. Zhong *et al.* showed the second highest frequency. Levin *et al.*, Pan *et al.*, and Perrone *et al.* showed the zero of the frequency. In the median, ours showed the largest value, which means the highest performance. Zhong *et al.* was the second. Levin *et al.* and Pan *et al.* showed the similar value, attaining the third and the fourth. Perrone *et al.* showed the smallest value. The performance was similar in the mean.

Table 6.13 SSIM of the latent images for DATA SET 3.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
1	1	0.2758	0.2181	0.6280	0.1294	**0.7382**
	2	0.3155	0.2360	0.5305	0.1026	**0.7342**
	3	0.3734	0.2316	0.6891	0.1349	**0.7762**
	4	0.2787	0.2355	0.6101	0.1032	**0.6580**
	5	0.3805	0.2535	0.6615	0.1592	**0.8255**
	6	0.3740	0.2574	0.6325	0.1363	**0.8044**
	7	0.2767	0.2567	0.6216	0.1359	**0.7912**
	8	0.3373	0.2406	0.6117	0.1149	**0.7560**
2	1	0.2952	0.2759	0.5844	0.1410	**0.6692**
	2	0.3588	0.2506	0.4836	0.1213	**0.6427**
	3	0.3712	0.2633	0.6638	0.2028	**0.7076**
	4	0.3323	0.2181	0.5483	0.1009	**0.5957**
	5	0.4083	0.2968	0.6534	0.3643	**0.7399**
	6	0.4169	0.2995	0.6193	0.1623	**0.6993**
	7	0.3998	0.2711	0.6069	0.1381	**0.7200**
	8	0.3794	0.2573	0.5916	0.1275	**0.6928**
3	1	0.3410	0.2514	0.6402	0.1422	**0.7544**
	2	0.3512	0.2510	0.5256	0.1236	**0.7157**
	3	0.3883	0.2711	0.7103	0.1904	**0.8148**
	4	0.3288	0.2556	0.6102	0.1161	**0.6614**
	5	0.4272	0.2814	0.6777	0.1832	**0.8223**
	6	0.3831	0.2720	0.6322	0.1296	**0.8158**
	7	0.3447	0.2904	0.6497	0.1319	**0.7949**
	8	0.3811	0.2615	0.6258	0.1309	**0.7322**
4	1	0.3016	0.2035	0.6246	0.0953	**0.7515**
	2	0.3029	0.1576	0.4933	0.0753	**0.6867**
	3	0.3435	0.2354	0.7127	0.1214	**0.8334**
	4	0.2863	0.1202	0.5289	0.0775	**0.5450**
	5	0.3573	0.2466	0.6658	0.1230	**0.8296**
	6	0.3511	0.2095	0.6424	0.0966	**0.7732**
	7	0.3078	0.1936	0.6293	0.0972	**0.6941**
	8	0.3637	0.1918	0.5931	0.0781	**0.6307**
#		0	0	0	0	32

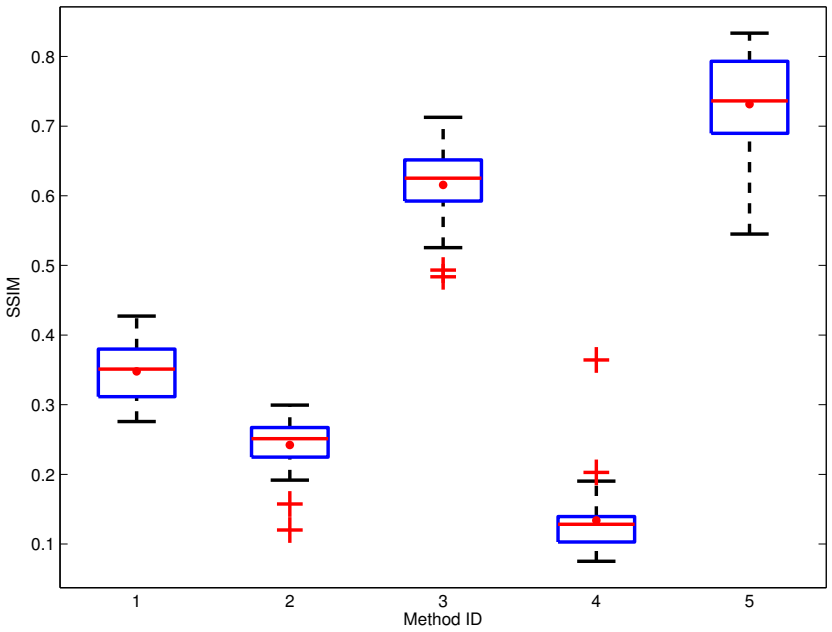


Figure 6.23 Box plot of SSIM of the latent images for DATA SET 3. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSIM for DATA SET 3, ours attained the highest performance, as shown in Table 6.13 and in Figure 6.23. In the number of the highest performance for each image, ours showed the highest performance for all of the images. In the median, ours showed the largest value, which means the highest performance. Zhong *et al.* was the second. Levin *et al.* was the third. Pan *et al.* was the fourth. Perrone *et al.* showed the smallest value. The performance was similar in the mean.

Table 6.14 SSD of the kernels for DATA SET 4.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	0.0146	0.0068	0.0065	0.0142	**0.0012**
	2	0.0138	0.0069	0.0145	0.0186	**0.0010**
	3	0.0165	0.0047	0.0044	0.0133	**0.0006**
	4	0.0185	0.0181	0.0157	**0.0113**	0.0170
	5	0.0217	0.0032	0.0078	0.1215	**0.0013**
	6	0.0316	0.0296	0.0491	0.0414	**0.0280**
	7	0.0250	0.0318	**0.0150**	0.0423	0.0337
	8	0.0234	0.0260	**0.0168**	0.0169	0.0268
6	1	0.0273	0.0312	0.0274	0.0223	**0.0194**
	2	0.0267	0.0654	0.0456	**0.0161**	0.0341
	3	0.0292	0.0646	0.0244	**0.0212**	0.0222
	4	0.0177	0.0577	0.0140	**0.0095**	0.0281
	5	0.0465	0.0310	0.0239	**0.0234**	0.0239
	6	0.0241	0.0719	0.0317	**0.0232**	0.0584
	7	0.0244	0.0862	0.0240	**0.0161**	0.0196
	8	0.0209	0.0464	0.0188	**0.0130**	0.0270
7	1	0.0225	0.0160	0.0095	0.0166	**0.0050**
	2	0.0185	0.0127	0.0140	0.0153	**0.0085**
	3	0.0299	0.0155	**0.0055**	0.0187	0.0105
	4	0.0174	0.0284	0.0105	**0.0100**	0.0205
	5	0.0246	0.0118	0.0065	0.0234	**0.0057**
	6	0.0307	**0.0163**	0.0464	0.0201	0.0388
	7	0.0224	0.0391	**0.0117**	0.0132	0.0392
	8	0.0171	0.0109	**0.0046**	0.0108	0.0271
8	1	0.0153	0.0106	0.0073	0.0180	**0.0062**
	2	0.0154	0.0127	0.0247	0.0145	**0.0028**
	3	0.0208	0.0149	**0.0049**	0.0300	0.0052
	4	0.0132	0.0302	**0.0050**	0.0085	0.0149
	5	0.0214	0.0126	0.0058	0.0933	**0.0043**
	6	0.0195	0.0139	0.0388	0.0169	**0.0078**
	7	0.0154	0.0421	0.0346	0.0100	**0.0057**
	8	0.0143	0.0110	0.0054	0.0124	**0.0034**
#		0	1	7	9	15

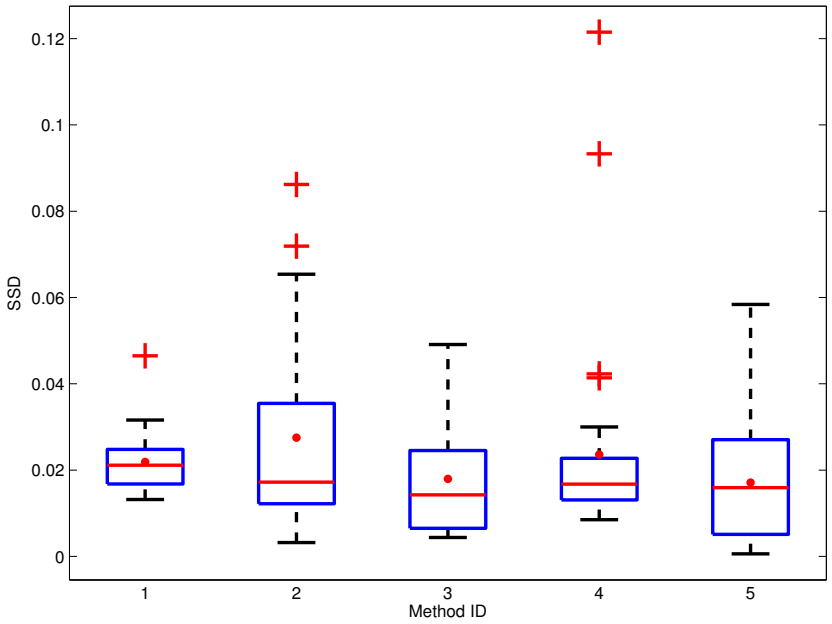


Figure 6.24 Box plot of SSD of the kernels for DATA SET 4. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the kernel estimation measured by SSD for DATA SET 4, ours attained the highest performance in the number of the highest performance for each image, and Zhong *et al.* attained the highest performance in the median and in the mean, as shown in Table 6.14 and in Figure 6.24. In the number of the highest performance for each image, ours showed the highest frequency, which means the highest performance. Perrone *et al.* was the second. Zhong *et al.* was the third. Pan *et al.* was the fourth. Levin *et al.* showed the lowest frequency. In the median, all of the methods showed the similar values. Among them, Zhong *et al.* showed the smallest value, which means the highest performance. The performance was similar in the mean.

Table 6.15 SSD of the latent images for DATA SET 4.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	854.9580	780.9134	418.0956	1747.3993	**146.9828**
	2	868.1719	784.3961	812.3791	2445.6004	**143.1998**
	3	797.7282	610.3369	144.4299	2521.1564	**112.2359**
	4	1746.4294	4130.2884	899.9390	3750.5574	**459.3012**
	5	627.2439	425.9840	149.4221	499.2249	**78.9084**
	6	1605.5730	1019.8693	697.5788	1799.8683	**121.1564**
	7	1292.7146	1747.9944	1053.0033	1950.8920	**981.4387**
	8	5700.7005	3249.5794	**1060.0563**	3299.2358	2101.3358
6	1	776.2589	428.9465	177.2015	2452.0491	**82.3201**
	2	635.6291	392.7028	119.5165	2412.7180	**66.7388**
	3	625.2986	313.1260	134.5024	1524.4144	**54.2915**
	4	777.8153	551.5315	312.9097	2887.8643	**83.0901**
	5	468.8552	370.0419	86.1342	1039.2049	**43.9514**
	6	794.5872	424.5637	415.8499	2176.0375	**51.8155**
	7	557.4702	333.0895	306.7848	2756.4251	**51.9495**
	8	529.2691	448.6843	278.4347	3066.2825	**65.7142**
7	1	1193.0013	735.8384	156.8440	3082.4614	**119.3567**
	2	930.2683	800.1798	175.3734	2776.4955	**148.0146**
	3	882.7247	554.7899	**94.0983**	1977.6709	94.3131
	4	1251.3330	856.0195	236.6076	3832.6707	**202.4586**
	5	621.8341	435.8570	**68.1188**	1609.9163	74.6465
	6	1318.7877	761.9423	**182.2082**	2528.8724	183.2349
	7	1291.9417	613.9905	**299.7379**	3344.0887	391.2838
	8	1277.2850	648.9143	154.8932	2779.6901	**103.0792**
8	1	735.2423	631.2222	193.3436	1829.4320	**164.4936**
	2	707.4536	708.3220	299.8952	3117.7720	**135.6835**
	3	686.1650	627.5722	126.5328	1348.9313	**113.4345**
	4	1412.6701	874.4029	249.6802	4075.4113	**182.5618**
	5	637.0340	453.8321	100.9613	423.3013	**95.5063**
	6	1566.1670	632.5090	147.9799	3171.9503	**99.9116**
	7	1264.6082	705.0136	**151.3140**	3682.9047	208.9924
	8	1188.4692	725.5125	246.8280	2681.9028	**136.7179**
#		0	0	6	0	26

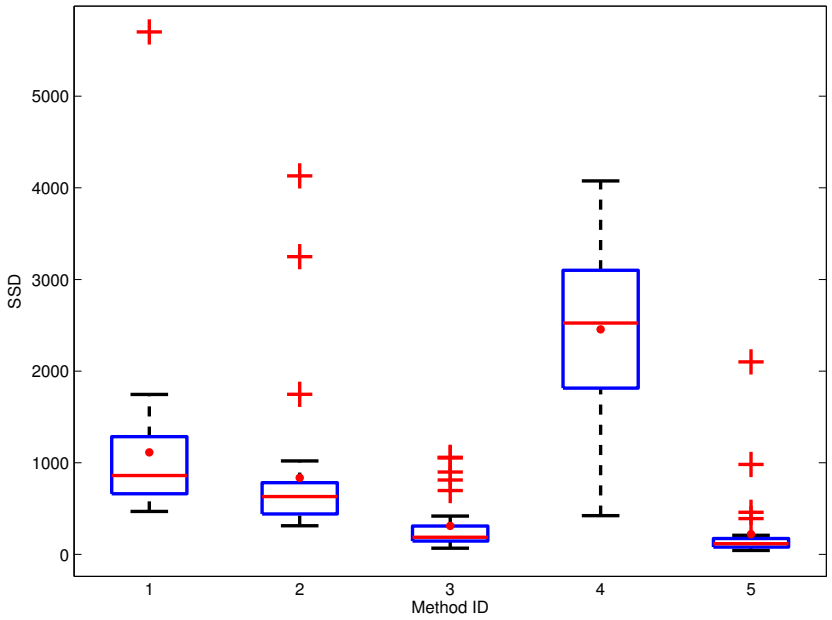


Figure 6.25 Box plot of SSD of the latent images for DATA SET 4. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSD for DATA SET 4, ours attained the highest performance, as shown in Table 6.15 and in Figure 6.25. In the number of the highest performance for each image, ours showed the highest performance for almost all of the images. Zhong *et al.* showed the second highest frequency. Levin *et al.*, Pan *et al.*, and Perrone *et al.* showed the zero of the frequency. In the median, Zhong *et al.* and ours showed the similarly lowest value, which means the high performance. Pan *et al.* was the third. Levin *et al.* was the fourth. Perrone *et al.* showed the largest value, which means the lowest performance. The performance was similar in the mean.

Table 6.16 PSNR of the latent images for DATA SET 4.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	17.67	16.72	21.43	13.29	**24.85**
	2	17.51	16.98	18.75	12.19	**24.88**
	3	18.11	17.40	25.54	11.83	**26.26**
	4	15.51	11.57	18.39	10.44	**20.79**
	5	18.78	18.93	25.23	19.18	**27.89**
	6	15.76	16.44	19.31	13.44	**25.90**
	7	16.58	14.64	17.72	13.10	**17.98**
	8	10.51	12.34	**17.69**	11.00	14.78
6	1	17.71	18.36	24.63	11.62	**26.63**
	2	18.33	18.90	25.45	11.61	**27.63**
	3	18.46	19.76	25.69	13.29	**28.67**
	4	17.96	17.62	22.36	11.09	**26.75**
	5	19.27	19.02	27.10	14.78	**29.73**
	6	17.62	18.57	21.09	11.99	**28.92**
	7	18.90	19.68	22.35	11.15	**28.95**
	8	19.19	18.40	22.77	10.82	**27.92**
7	1	16.33	16.74	25.22	11.12	**25.55**
	2	17.21	16.50	24.57	11.42	**24.85**
	3	17.42	17.73	**27.24**	12.70	27.10
	4	16.17	16.26	23.33	10.37	**23.71**
	5	18.50	18.59	28.07	13.42	**28.26**
	6	15.88	16.80	24.36	11.82	**24.43**
	7	15.97	17.51	**22.45**	10.83	21.39
	8	16.13	17.10	25.08	11.39	**26.42**
8	1	18.14	17.17	**24.77**	13.04	24.73
	2	18.40	16.81	22.61	11.12	**25.35**
	3	18.36	17.15	26.35	14.35	**26.50**
	4	16.19	16.36	23.83	10.16	**24.36**
	5	18.73	18.42	26.82	19.07	**27.40**
	6	15.83	17.35	25.57	11.04	**26.87**
	7	16.64	17.09	**25.51**	10.47	24.38
	8	16.83	16.92	23.82	11.66	**25.68**
#		0	0	5	0	27

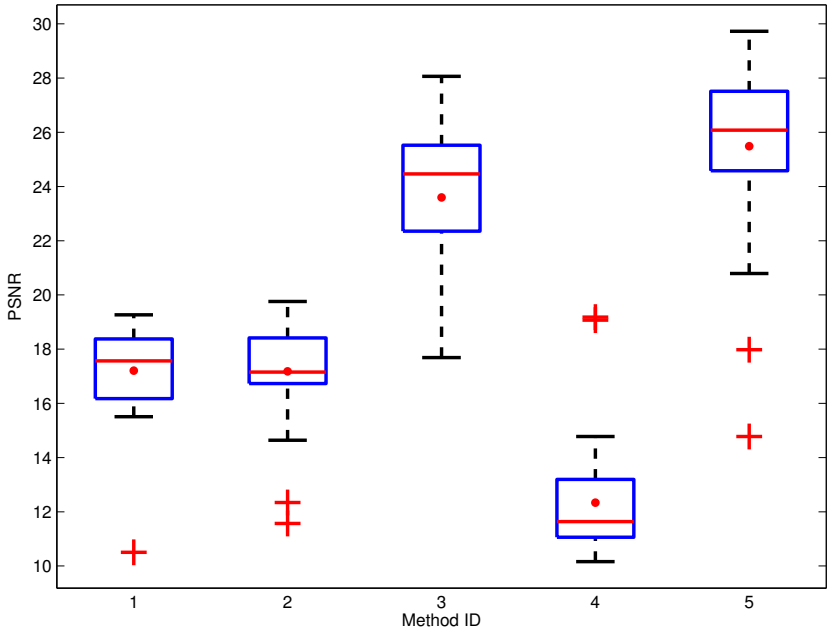


Figure 6.26 Box plot of PSNR of the latent images for DATA SET 4. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by PSNR for DATA SET 4, ours attained the highest performance, as shown in Table 6.16 and in Figure 6.26. In the number of the highest performance for each image, ours showed the highest performance for almost all of the images. Zhong *et al.* showed the second highest frequency. Levin *et al.*, Pan *et al.*, and Perrone *et al.* showed the zero of the frequency. In the median, ours showed the largest value, which means the highest performance. Zhong *et al.* was the second. Levin *et al.* and Pan *et al.* showed the similar value, attaining the third and the fourth. Perrone *et al.* showed the smallest value. The performance was similar in the mean.

Table 6.17 SSIM of the latent images for DATA SET 4.

im.	ker.	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
5	1	0.4729	0.3880	0.6346	0.2673	**0.7374**
	2	0.4425	0.4055	0.4983	0.1880	**0.7437**
	3	0.5009	0.4151	0.7679	0.2431	**0.7900**
	4	0.4556	0.1653	0.5517	0.2086	**0.6327**
	5	0.5139	0.4718	0.7390	0.4372	**0.8465**
	6	0.4672	0.4111	0.5522	0.2688	**0.7859**
	7	0.4913	0.2747	**0.5567**	0.2676	0.5275
	8	0.2191	0.0994	**0.5300**	0.2108	0.2722
6	1	0.2325	0.1729	**0.5813**	0.0411	0.5449
	2	0.2544	0.1679	0.4486	0.0393	**0.5861**
	3	0.2718	0.1911	0.6126	0.0699	**0.6232**
	4	0.2963	0.1286	0.5137	0.0450	**0.5568**
	5	0.2746	0.1942	0.6056	0.0935	**0.6875**
	6	0.2985	0.1687	0.4467	0.0571	**0.6647**
	7	0.2947	0.1958	0.4987	0.0563	**0.6672**
	8	0.3070	0.1648	0.5242	0.0488	**0.6191**
7	1	0.3035	0.1860	**0.7302**	0.0881	0.6120
	2	0.3191	0.1945	0.6045	0.0858	**0.6129**
	3	0.3070	0.2196	**0.7421**	0.1172	0.6939
	4	0.3462	0.1949	**0.7105**	0.0846	0.5838
	5	0.3271	0.2493	0.7006	0.1201	**0.7557**
	6	0.3048	0.2098	0.6087	0.1088	**0.6792**
	7	0.3049	0.2336	**0.6616**	0.1017	0.5882
	8	0.3266	0.2162	0.6699	0.0905	**0.6962**
8	1	0.4170	0.3071	**0.7097**	0.1638	0.6434
	2	0.4402	0.2938	0.5186	0.1396	**0.6779**
	3	0.4364	0.2910	**0.7444**	0.1987	0.7237
	4	0.4083	0.2768	**0.7099**	0.1257	0.6467
	5	0.4551	0.3485	0.7368	0.3317	**0.7776**
	6	0.4315	0.3489	0.7122	0.1738	**0.7512**
	7	0.4705	0.3304	**0.7259**	0.1492	0.7120
	8	0.4297	0.3242	0.7079	0.1430	**0.7210**
#		0	0	11	0	21

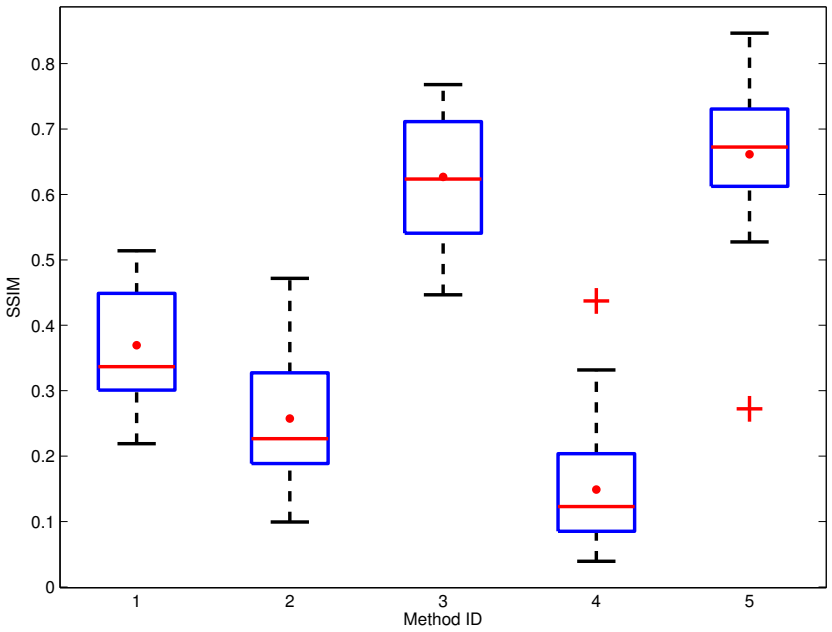


Figure 6.27 Box plot of SSIM of the latent images for DATA SET 4. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

In the accuracy on the image estimation measured by SSIM for DATA SET 4, ours attained the highest performance, as shown in Table 6.17 and in Figure 6.27. In the number of the highest performance for each image, for all of the images, ours showed the highest performance. In the median, ours showed the largest value, which means the highest performance. Zhong *et al.* was the second. Levin *et al.* and Pan *et al.* showed the similar value, attaining the third and the fourth. Perrone *et al.* showed the smallest value. The performance was similar in the mean.

6.2 Images without Ground Truth

We also conducted experiments on images without ground truth. We examined the visual quality of the deblurred images qualitatively. Also, We quantitatively evaluated those images using various metrics from the field of image quality assessment. Those images are of two kinds one of which contains blur only, and the other contains noise as well. The results of images with blur only are listed in Figures 6.28 and 6.29, those with blur and noise are in Figures 6.30 and 6.31. In each figures, we listed the deblurred images and the estimated blur-kernels on the first row. On the second row, we listed the part of the image which is marked with a white rectangle in the blurred input image.

In Figure 6.28, we listed the results for the image of many buildings. The result of ours shows the most uniform color for the areas without variation. The result of Zhong *et al.* [88] looks still blurry, and that of Perrone *et al.* [61] shows stair-casing effects. In the result of Pan *et al.* [58], the blue and the red spread out. The result of Levin *et al.* [46] is still slightly more blurry than ours. In Figure 6.29, we listed the results for a classical building. The results of Pan *et al.* [58] and ours are very clean and look similar for both of the estimated kernels and images. The kernels of Levin *et al.* [46] and Zhong *et al.* [88] contain unexpected motion. Thus, the estimated images contain a slight blur.

In Figure 6.30, we listed the results for the images of a shop with blur and noise. As shown in the figure, the letters are difficult to be recognized for the results of Levin *et al.* [46] and Pan *et al.* [58]. The result of Perrone *et al.* [61] still contains noise. The results of Zhong *et al.* [88] and ours show the highest quality. In Figure 6.31, we listed the results for the images of a house with blur and noise. The results of Levin *et al.*, Pan *et al.* [58], and Perrone *et al.* [61] are still blurry and noisy. The blur and noise are removed well for the results of Zhong *et al.* [88] and ours; ours more clearly deblurred the frame of the window and the structure of the bricks.

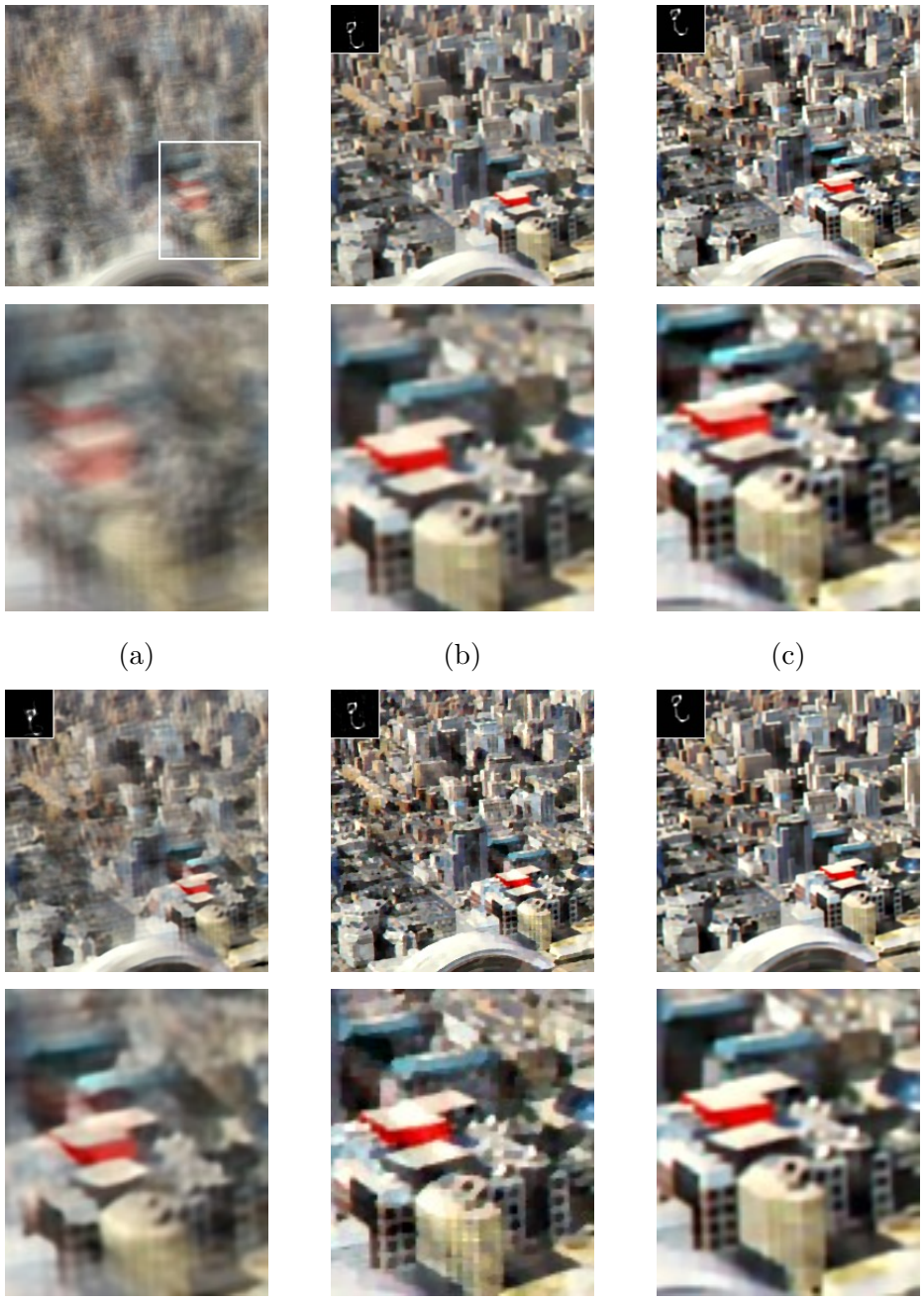


Figure 6.28 Deblurred results of a blurry image without ground truth. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

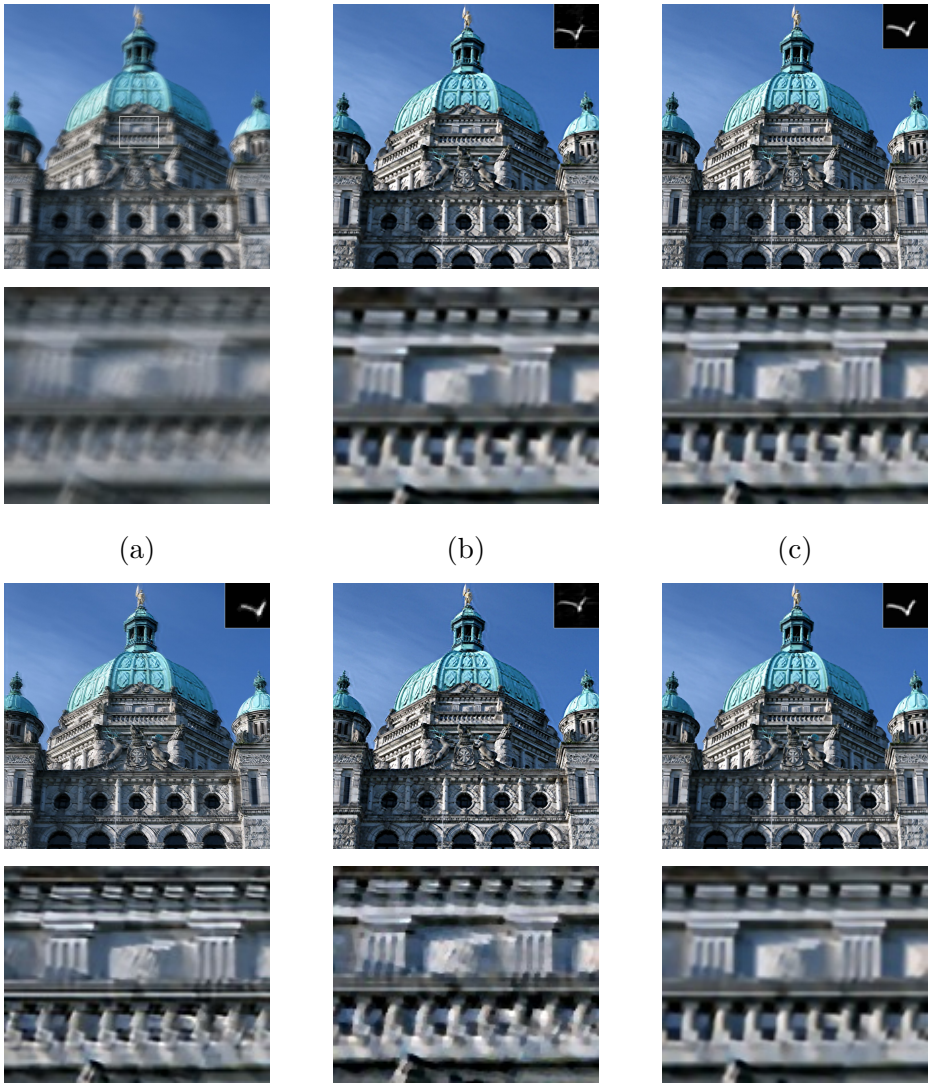
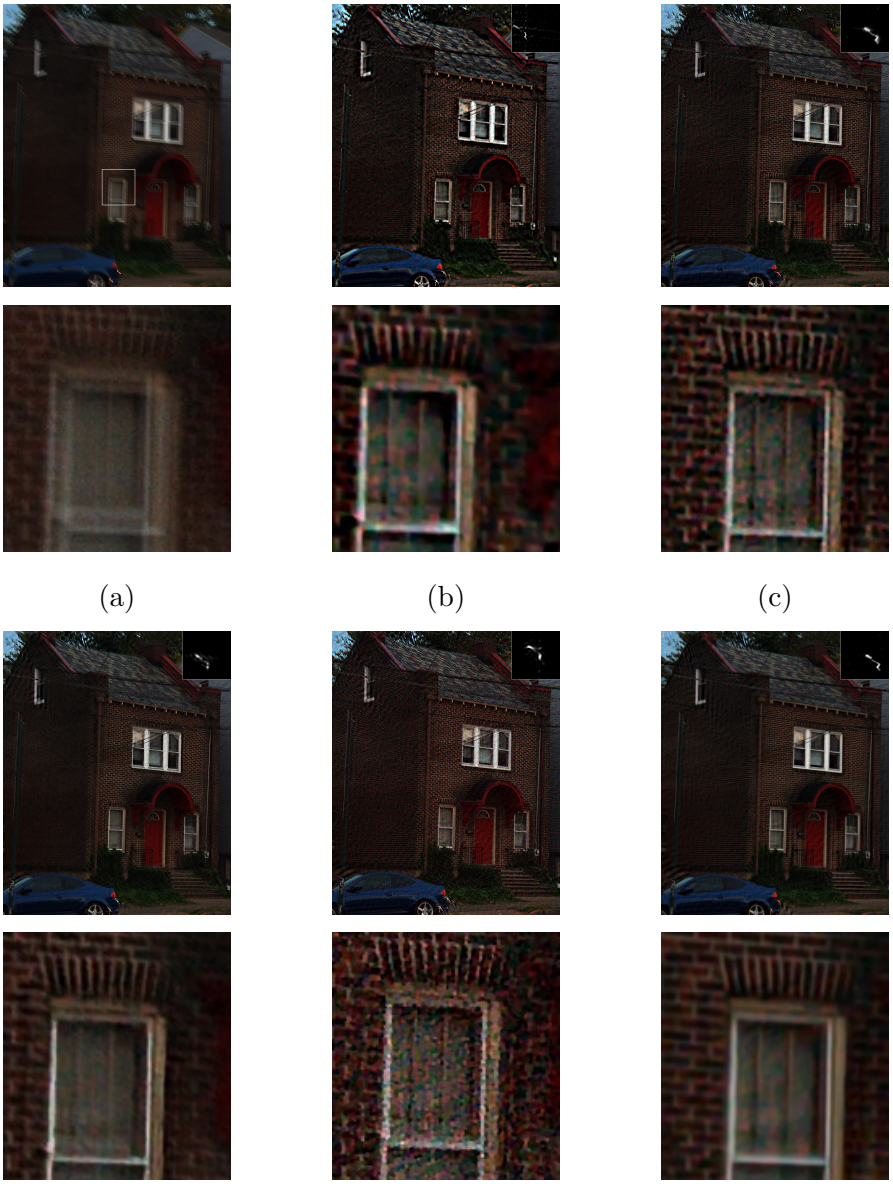


Figure 6.29 Deblurred results of a blurry image without ground truth. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

(a) (b) (c)

(d) (e) (f)

Figure 6.30 Deblurred results of a blurry and noisy image without ground truth. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

Figure 6.31 Deblurred results of a blurry and noisy image without ground truth. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

Table 6.18 No reference image quality assessments of the images in Fig. 6.28.

Metric	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
SSEQ	42.3056 (4)	**44.2900** (1)	43.2367 (2)	27.1332 (5)	42.4936 (3)
BRISQUE	103.9250 (2)	95.4194 (3)	**113.3570** (1)	88.1143 (5)	94.5288 (4)
STA6	7.5909 (4)	8.1071 (3)	6.8903 (5)	**8.4438** (1)	8.1416 (2)
DCT1	2.6315 (4)	2.9621 (3)	2.1320 (5)	**3.0419** (1)	3.0281 (2)

Table 6.19 No reference image quality assessments of the images in Fig. 6.29.

Metric	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
SSEQ	**38.0969** (1)	34.5328 (3)	25.2456 (4)	24.1913 (5)	35.2264 (2)
BRISQUE	**106.2890** (1)	103.9100 (2)	103.3200 (4)	101.6330 (5)	103.7910 (3)
STA6	9.6365 (5)	9.8840 (4)	**10.4268** (1)	10.3635 (2)	9.8910 (3)
DCT1	5.4015 (4)	5.4160 (3)	5.3148 (5)	**5.6538** (1)	5.4526 (2)

Table 6.20 No reference image quality assessments of the images in Fig. 6.30.

Metric	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
SSEQ	**42.7512** (1)	32.8993 (3)	27.9763 (4)	8.2577 (5)	41.9484 (2)
BRISQUE	117.2360 (2)	114.8650 (4)	117.1040 (3)	108.8030 (5)	**118.7530** (1)
STA6	8.9521 (3)	9.4186 (2)	8.6291 (4)	**9.6929** (1)	8.1425 (5)
DCT1	6.0000 (3)	**6.1596** (1)	5.1603 (5)	6.0790 (2)	5.5303 (4)

Table 6.21 No reference image quality assessments of the images in Fig. 6.31.

Metric	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
SSEQ	40.9144 (4)	43.0987 (2)	41.0107 (3)	25.4025 (5)	**49.3301** (1)
BRISQUE	116.5990 (3)	113.8740 (4)	**118.1990** (1)	105.3770 (5)	117.3700 (2)
STA6	9.3384 (3)	9.4552 (2)	8.9423 (4)	**10.1236** (1)	8.4252 (5)
DCT1	**6.6729** (1)	6.3281 (3)	5.7991 (4)	6.5920 (2)	5.7514 (5)

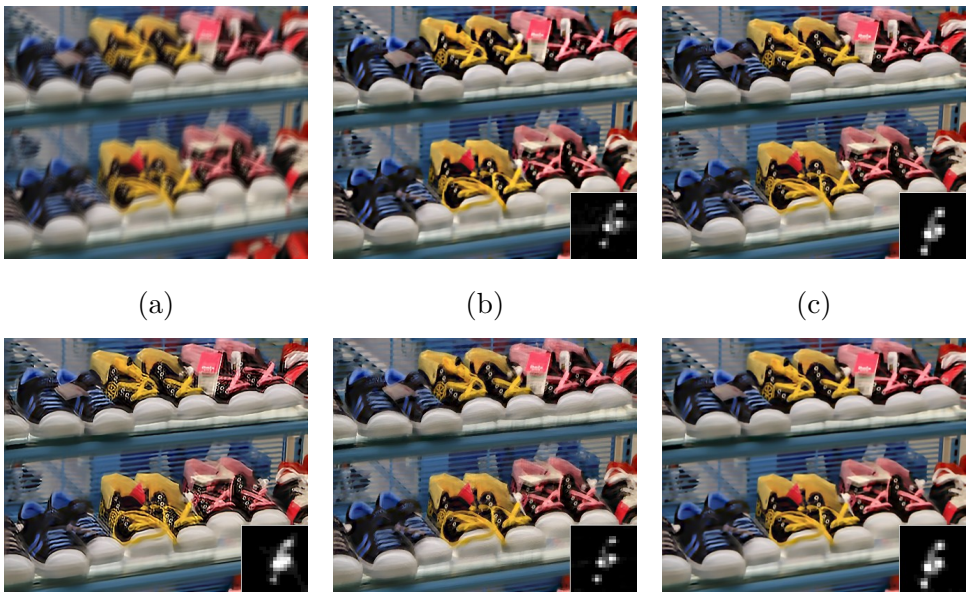
To quantitatively evaluate the quality of the deblurred images in Figures 6.28, 6.29, 6.30, and 6.31, we computed various metrics and listed the values in Tables 6.18, 6.19, 6.20, and 6.21, respectively. The metrics are dubbed spatial-spectral entropy-based quality index (SSEQ) [47], dubbed blind/referenceless image spatial quality evaluator (BRISQUE) [53], modified gray-level variance (STA6) [62], and discrete cosine transform energy ratio (DCT1) [62]. The SSEQ and BRISQUE are learning-based metrics, and the STA6 and DCT1 are hand-crafted metrics. As shown in Tables 6.18 and 6.19, Ours showed moderate results. In Table 6.18, Perrone *et al.* attained the highest performance for the hand-crafted metrics but the lowest for the learning-based metrics. In Table 6.19, Levin *et al.* attained the highest performance for the learning-based metrics but very low performance for the hand-crafted ones. For images with blur and noise, ours attained very high performance for the learning-based metrics but very low for the hand-crafted ones as shown in Tables 6.20 and 6.21. The results of Perrone *et al.* took the opposite to ours. They attained very high performance for the hand-crafted metrics but low for the learning-based metrics.

6.3 Analysis on Preprocessing using Denoising

We conducted experiments to examine the effect of the denoising method as a preprocessing step. We performed image deblurring to the images without noise, those with noise, and those denoised by various methods. For denoising, Gaussian filtering, bilateral filtering, nonlocal means filtering, and the BM3D method are used.

First, we performed deblurring to shoes image without noise and listed the results in Figure 6.32. As shown in the figure, most of the methods showed quite good results for the images without noise. In Figure 6.33, we listed the deblurred results on the images with noise using various methods without preprocessing of denoising. For the noisy image, only Zhong *et al.* and ours showed good re-

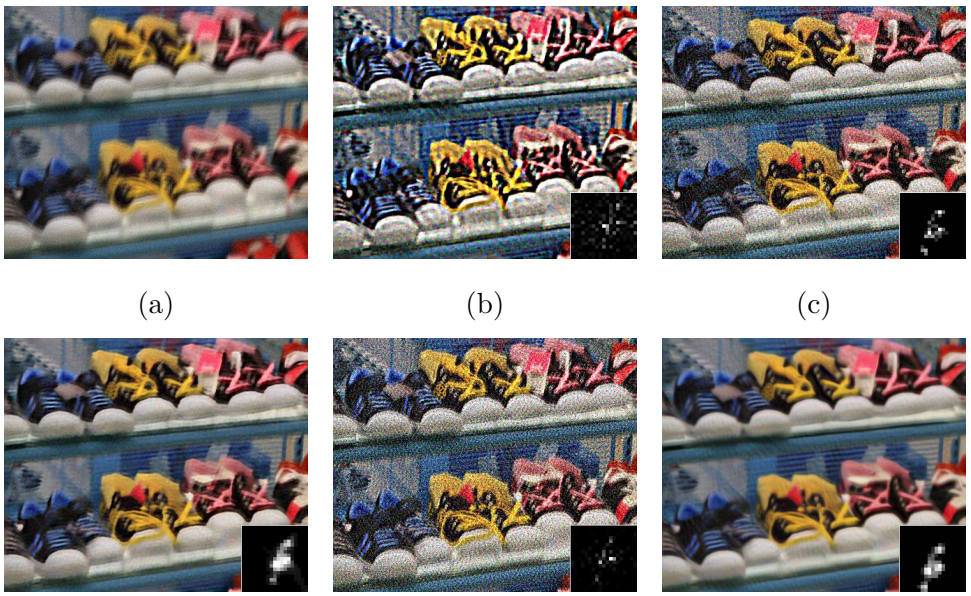
sults. The results of Levin *et al.*, Pan *et al.*, and Perrone *et al.* contains much noise. The images in Figure 6.34 are the deblurred results using preprocessing of Gaussian denoising. All of the results showed very blurry results. The image denoised by Gaussian filtering shows too blurry and lost much of edge information. Therefore, all of the deblurring methods could not reconstruct the sharp edges. The images in Figure 6.35 are the deblurred results using preprocessing of bilateral filtering. The denoised image maintained some edges at the boundary of the shoes. Thus, the results are better than when the Gaussian filtering was applied. Nevertheless, Levin *et al.*, Pan *et al.*, and Perrone *et al.* could not estimate the correct kernel, thus resulting still blurry images. Zhong *et al.* and ours showed better results than others, but they are still blurry. The images in Figure 6.36 are the deblurred results using preprocessing of nonlocal means denoising. All of the methods except Perrone *et al.* seem to approximately estimate the ground truth kernel. The images in Figure 6.37 are the deblurred results using preprocessing of BM3D denoising. The estimated kernels are a little blurry than those using nonlocal means denoising, but the results contain less noise.



(a) (b) (c)

(d) (e) (f)

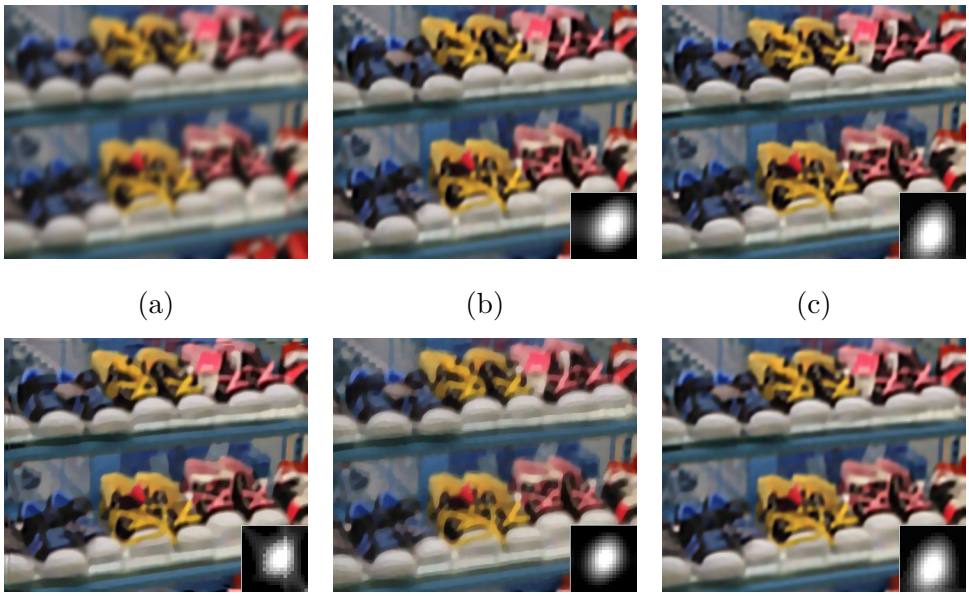
Figure 6.32 The results of each method on the image of shoes without noise. (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

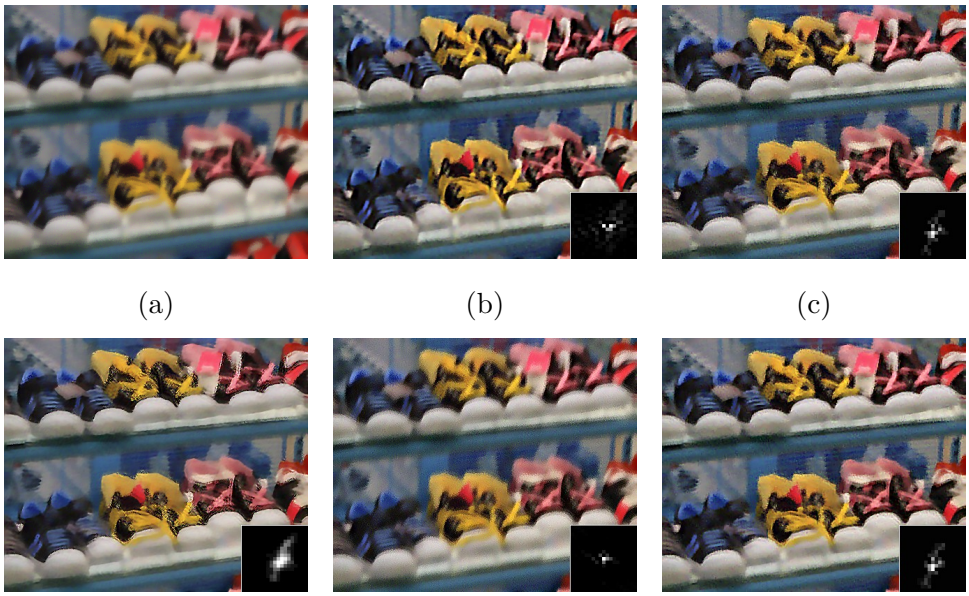
Figure 6.33 The results of each method on the image of shoes with noise without preprocessing (a) Blurred image. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

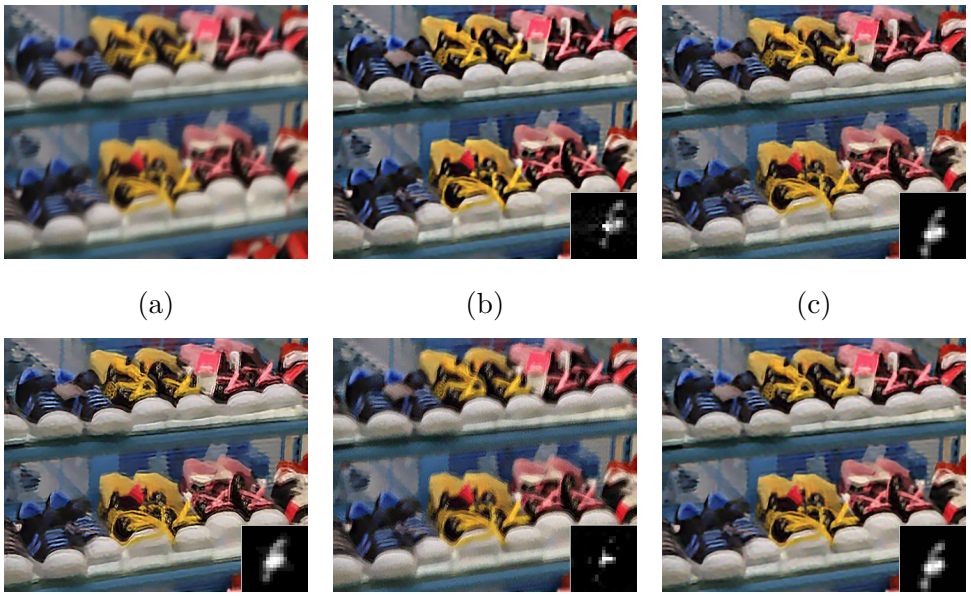
Figure 6.34 The results of each method on the image of shoes using Gaussian filtering as preprocessing. (a) Blurred image which is denoised by Gaussian filtering. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

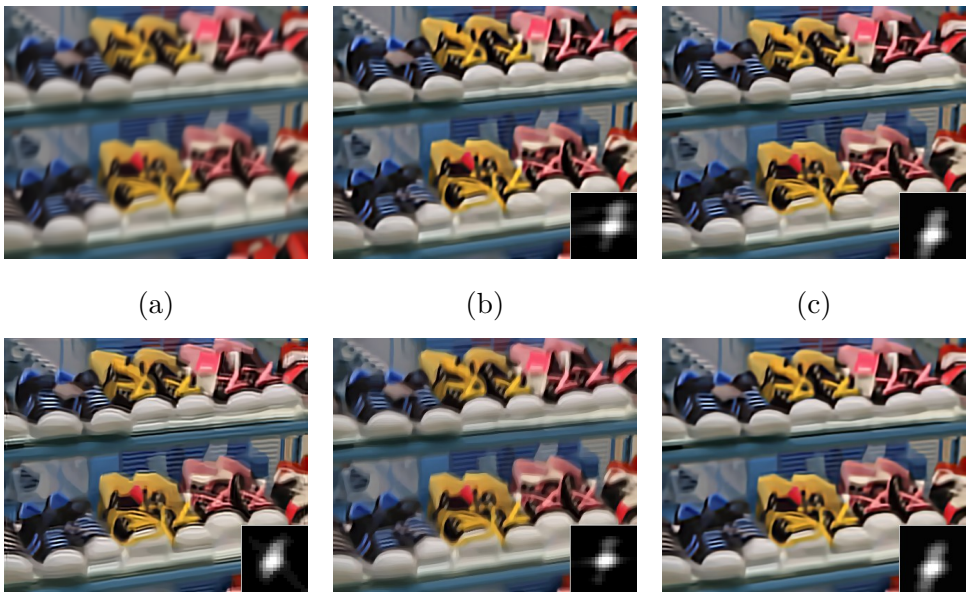
Figure 6.35 The results of each method on the image of shoes using bilateral filtering as preprocessing. (a) Blurred image which is denoised by bilateral filtering. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

Figure 6.36 The results of each method on the image of shoes using nonlocal means denoising as preprocessing. (a) Blurred image which is denoised by nonlocal means denoising. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.



(a) (b) (c)

(d) (e) (f)

Figure 6.37 The results of each method on the image of shoes using BM3D as preprocessing. (a) Blurred image which is denoised by BM3D. (b) Levin *et al.* [46]. (c) Pan *et al.* [58]. (d) Zhong *et al.* [88]. (e) Perrone *et al.* [61]. (f) Ours.

Table 6.22 SSD of the kernels for the shoes image.

Image Type	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
no noise	0.0044	0.0010	0.0045	0.0036	**0.0008**
no denoising	0.0166	0.0088	0.0044	0.0205	**0.0011**
denoising by Gaussian filter	0.0147	0.0146	0.0144	**0.0133**	0.0145
denoising by bilateral filter	0.0269	0.0164	**0.0099**	0.0955	0.0137
denoising by nonlocal means	0.0061	**0.0039**	0.0071	0.0292	**0.0039**
denoising by BM3D	0.0103	**0.0086**	0.0100	0.0109	**0.0086**

Table 6.23 SSD of the latent images for the shoes image.

Image Type	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
no noise	379.1797	157.9652	735.5135	323.0553	**125.8750**
no denoising	3018.2255	1607.8594	559.4924	3390.0566	**526.4957**
denoising by Gaussian filter	1198.3839	1061.8397	**1031.4703**	1308.4441	1058.1214
denoising by bilateral filter	957.1507	914.2478	1180.7939	1389.1224	**827.6636**
denoising by nonlocal means	596.1385	487.5835	755.5187	954.2882	**453.7089**
denoising by BM3D	781.4676	514.2906	841.5677	884.3208	**497.9316**

To quantitatively evaluate, we computed the SSD of the kernels and the estimated image; and PSNR and SSIM of the image. The results are listed in Tables 6.22, 6.23, 6.24, and 6.25. As shown in Table 6.22, ours showed the smallest values for the image without noise, that with noise but no denoising, that denoised by nonlocal means, and that denoised by BM3D. About SSD of the latent image, as shown in Table 6.23, ours showed the smallest values for all test cases except that denoised by Gaussian filtering. About PSNR of the latent image, as shown in Table 6.24, ours showed the largest values for all test cases except that denoised by Gaussian filtering. as shown in Table 6.25, ours showed the largest values for all test cases except those denoised by Gaussian filtering and bilateral filtering. Preprocessing of nonlocal means denoising improved the

Table 6.24 PSNR of the latent images for the shoes image.

Image Type	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
no noise	25.68	29.41	23.04	26.30	**30.32**
no denoising	16.39	16.85	23.64	13.95	**24.02**
denoising by Gaussian filter	20.72	21.22	**21.36**	20.30	21.24
denoising by bilateral filter	21.54	21.58	20.72	19.80	**22.04**
denoising by nonlocal means	23.31	23.87	22.14	21.34	**24.16**
denoising by BM3D	22.46	24.08	22.15	21.88	**24.18**

Table 6.25 SSIM of the latent images for the shoes image.

Image Type	Levin *et al.*	Pan *et al.*	Zhong *et al.*	Perrone *et al.*	Ours
no noise	0.8432	0.9143	0.7995	0.8570	**0.9280**
no denoising	0.4394	0.3297	0.7003	0.2113	**0.7428**
denoising by Gaussian filter	0.6028	0.6243	**0.6349**	0.5757	0.6246
denoising by bilateral filter	**0.6565**	0.6123	0.6334	0.4708	0.6511
denoising by nonlocal means	0.7291	0.7237	0.6859	0.6078	**0.7349**
denoising by BM3D	0.7154	0.7661	0.6949	0.6793	**0.7733**

(a) (b) (c) (d) (e)

Figure 6.38 Images for testing the effect of size of nonlocal patches. First row: images with simple structure. Second row: images with complex structure. (a) Ground truth images with ground truth kernel. (b) Blurred images. (c) Deblurred images with patch size of 5×5. (d) Deblurred images with patch size of 9×9. (e) Deblurred images with patch size of 13×13.

image quality for all of the methods except Zhong *et al.* and ours. Preprocessing of BM3D denoising improved the image quality for all of the methods except Zhong *et al.* The improved quality of the others methods are still lower than ours without preprocessing. About PSNR and SSIM, ours with preprocssing of BM3D showed the highest image quality.

6.4 Analysis on the Size of Nonlocal Patches

We also conducted experiments for analyzing the effect of the size of nonlocal patches. If the size is small, many same patches may be found. However, if the size is too small, the information of the patches may be useless. On the other hand, if the size is too large, the similarity of the nonlocal patches may be decrease. In this case, the information of the patches may be harmful. Therefore, we expect that a certain size may be the most effective.

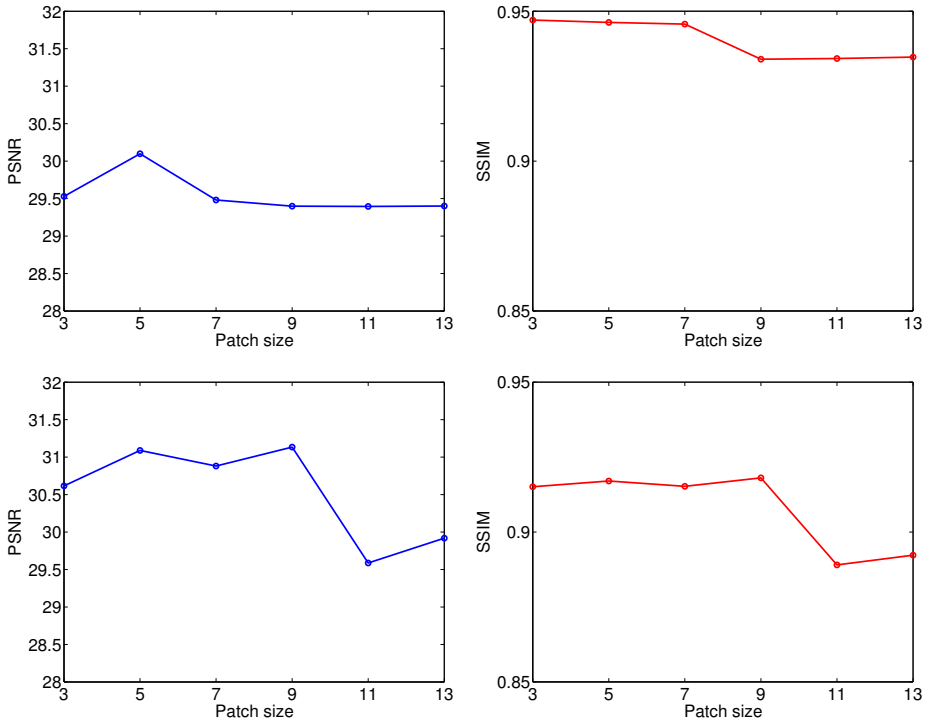
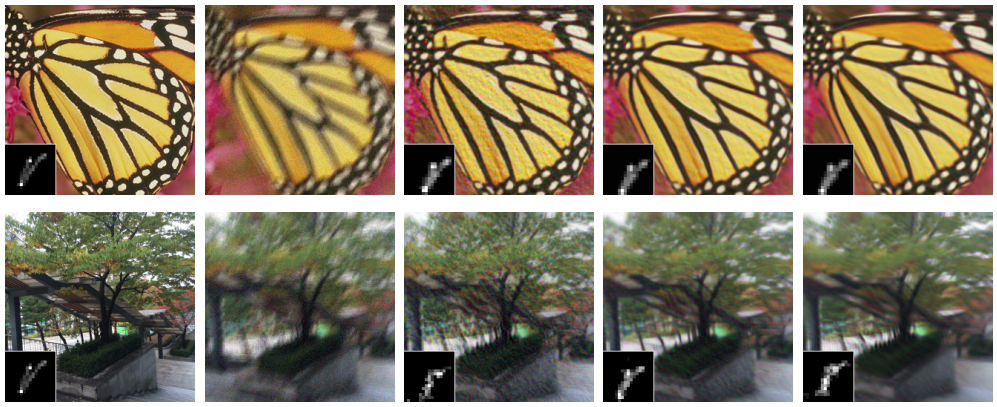


Figure 6.39 Change of PSNR and SSIM by varying the size of nonlocal patches. First row: results of the images on the first row in Fig. 6.38. Second row: results of the images on the second row in Fig. 6.38.

Depending on the characteristics of objects in images, the effective size may be different. For example, images with simple structure may maintain the same similarity even when the size is larger, but image with complex structure may not. Therefore, the effective size may be smaller in images with complex structure than in those with simple structure. In addition, the noise in images also affects the size. If images have noise, the same patches of the small size could be selected by chance. In that case, the nonlocal patches could not provide effective information. Therefore, the effective size may be larger in images with noise than in those without noise.

To verify our hypothesis, we conducted experiment on two types of images



(a) (b) (c) (d) (e)

Figure 6.40 Images for testing the effect of size of nonlocal patches. First row: images with simple structure. Second row: images with complex structure. (a) Ground truth images with ground truth kernel. (b) Blurred images. (c) Deblurred images with patch size of 5×5. (d) Deblurred images with patch size of 9×9. (e) Deblurred images with patch size of 13×13.

by varying the size of patches. One of the images has a simple structure, and the other has complex structure. These images are shown in Figure 6.38. In the figure, the image in the first row has a simple structure, and the image in the second row has a complex structure. When the visual quality is considered in Figure 6.38 (c), (d), and (e), all of the images shows similar and high quality. For quantitative evaluation, we computed the PSNR and SSIM and illustrated them in Figure 6.39. As shown in the figure, PSNR rose and then lowered for both of the images as the size became larger. SSIM also lowered around 7 or 9 of the size. This indicates that a certain size is more effective and a too large size is also ineffective.

We also performed the same experiment on the two images after adding noise to them. These images are shown in Figure 6.40. When the visual quality is considered in Figure 6.40 (c), (d), and (e), the visual quality in the first row

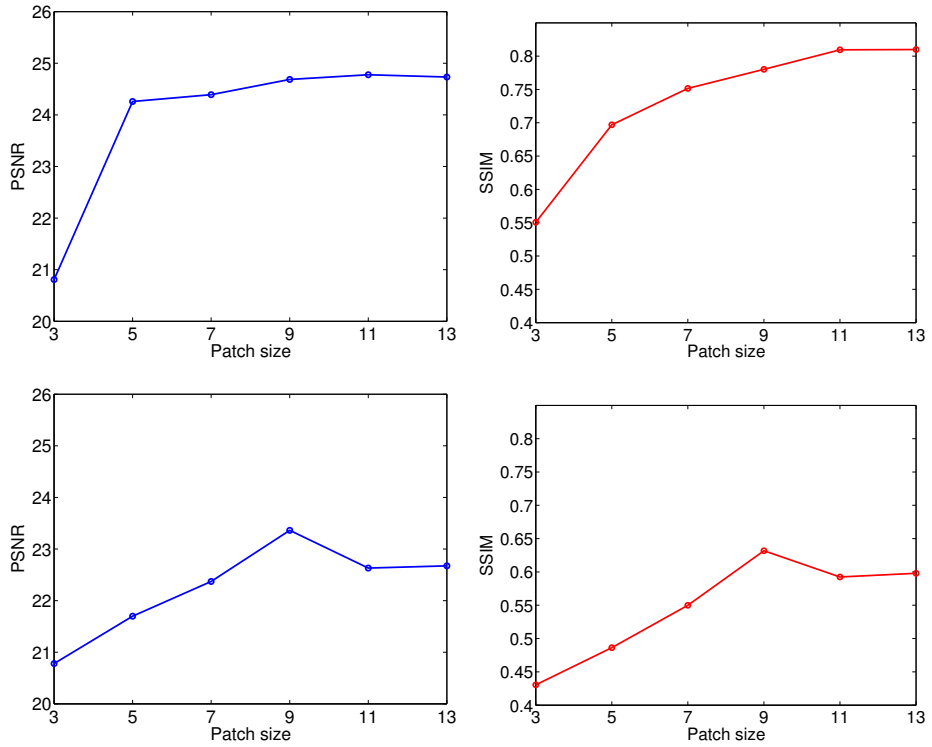


Figure 6.41 Change of PSNR and SSIM by varying the size of nonlocal patches. First row: results of the images on the first row in Fig. 6.40. Second row: results of the images on the second row in Fig. 6.40.

became better, and that in the second row became better and then worse as the size became larger. For quantitative evaluation, we computed the PSNR and SSIM and illustrated them in Figure 6.41. In the figure, the same phenomenon was observed. PSNR and SSIM went higher for the image in the first row, and those for the image in the second row went up and then went down as the size became larger. This indicates that effective size is smaller for images with complex structure than for those with simple structure. In addition, the effective size was larger in noisy images than in images without noise.

Table 6.26 System and data specification.

CPU	Intel i5-2500 3.3 GHz
RAM	8 GB
OS	Windows 7 Enterprise 64 bit
P.L.	MATLAB ver. 8.1
Data	32 examples of 256×256 size

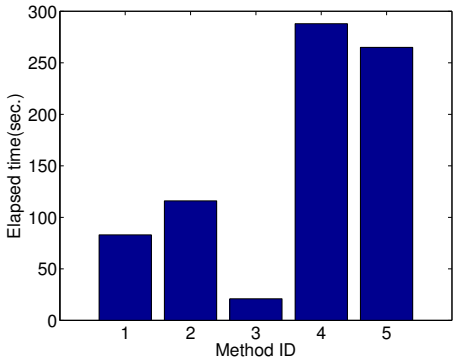


Figure 6.42 Time performance. Method ID 1: Levin *et al.* [46], Method ID 2: Pan *et al.* [58], Method ID 3: Zhong *et al.* [88], Method ID 4: Perrone *et al.* [61], and Method ID 5: Ours.

6.5 Time Performance

We evaluated and compared the time performance of the various methods of blind image deblurring. The system and data specification for this experiment is listed on Table 6.26. We used the DATA SET 1, which is composed of 32 images of 256×256 size, and the average time is illustrated in Figure 6.42. As shown in the figure, Zhong *et al.* was fastest, and Perrone *et al.* was slowest. Levin *et al.* and Pan *et al.* took moderate time around 100 sec. Ours consumed time of twice as much as that in Pan *et al.* This is due to the nonlocal operation.

Chapter 7

Conclusion

In this paper, we have proposed an effective method for blind image deblurring. Blind image deblurring aims to estimate the blur kernel and the latent image from a blurry image. Since this estimation is a severely ill-posed problem, effective regularizations are required for the successful estimation. For the regularization, methods of sparse prior and selective edges showed great success [31, 19, 46, 58]. However, there is still a room for improvement. The blind image deblurring usually consists of three estimation: kernel estimation, interim image estimation, and the latent image estimation. To improve each estimation, we proposed to use nonlocal patches using the self-similarity property of natural images.

First, for the blur kernel estimation, we proposed to use nonlocal patches selected by the similarity weighted by the kernel. The proposed method combined salient edges and nonlocal regularization. The salient edges provided reliable edge information, and the nonlocal prior offers data-authentic information. Ordinary nonlocal similar patches could have quite different values under a convolution operation because the convolution amplifies or reduces the effect of each element depending on the kernel. Therefore, we selected the nonlocal patches

by the similarity of patches weighted by a current estimate of the kernel.

Second, for the estimation of the interim image and the latent image, we proposed to use the nonlocal low-rank image as prior knowledge. This approach enables to handle the noise and to obtain high-quality image even for the blurred image with noise. We constructed a nonlocal low-rank image from a single image using the low-rank property of the nonlocal similar patches. Our low-rank image was composed of several low-rank patches, each of which was constructed by its own nonlocal patches. These nonlocal low-rank patches were obtained from the nonlocal patches by applying the singular value thresholding algorithm [13]. These nonlocal low-rank patches have less noise by its intrinsic of the low-rank property. Therefore, imposing this low-rank image as prior knowledge enabled to estimate high-quality images with less noise.

We conducted intensive experiments to evaluate the performance of our method and compared the results with those of conventional blind deblurring methods including Levin *et al.* [46], Pan *et al.* [58], Zhong *et al.* [88], and Perrone *et al.* [61]. Using ground truth images and ground truth kernels, we made blurry images and then deblurred them. We computed the sum of squared difference (SSD), the peak signal to noise ratio (PSNR), and the structural similarity index (SSIM). To evaluate the performance on blurry and noisy images, we also added white Gaussian noise to the blurry images, and computed the same metrics as well. The experimental results showed that our method is generally superior to the conventional method for the images with blur only. For the noisy and blurry images, our methods highly outperformed the conventional methods.

We also conducted experiments on the blurry images without ground truth. For the images with structural objects, results of our method showed sharper edges and smoother flat regions. For blurry and noisy images, only the method of Zhong *et al.* [88] and ours successfully removed the noise and blur, and the results of ours were clearer than those of Zhong *et al.* [88]. To quanti-

tatively evaluate the quality of those images without ground truth, we computed various metrics proposed by fields of no reference image quality assessment such as dubbed spatial-spectral entropy-based quality index (SSEQ) [47], dubbed blind/referenceless image spatial quality evaluator (BRISQUE) [53], modified gray-level variance (STA6) [62], and discrete cosine transform energy ratio (DCT1) [62]. Ours attained moderate performance for these metrics.

We analyzed the effect of the size of image patches. To examine this, we conducted experiment by varying the size of patches. The experimental results showed that the size about 7×7 was the most effective for images without noise and the size about 9×9 was the most effective for images with complex structure and noise. We also examined the time performance. We performed the blind image deblurring to 32 examples of size of 256×256 and averaged the elapsed time. The experimental results showed that ours consumes time of twice as much as conventional approaches such as Levin *et al.* and Pan *et al.* The more time consumption was due to the nonlocal operations.

In conclusion, although ours consumes much time than conventional methods, the proposed method is effective for ordinary blurry images and highly outperforms conventional methods for blurry images with noise.

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] A. Albert. *Regression and the Moore-Penrose pseudoinverse*. Elsevier, 1972.

[3] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and superresolution from an image sequence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 571–582. Springer, 1996.

[4] A. Ben-Israel and T. N. Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[5] A. Björck. *Numerical methods for least squares problems*. Siam, 1996.

[6] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Transactions on Image Processing*, 16(11):2675–2681, 2007.

[7] D. H. Brainard and W. T. Freeman. Bayesian color constancy. *JOSA A*, 14(7):1393–1411, 1997.

[8] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65. IEEE, 2005.

[10] C. Burrus. Vector space methods in signal and system theory. *Unpublished notes, ECE Dept., Rice University, Houston, TX*, 2012, 1994.

[11] C. Burrus and J. Barreto. Least p-power error design of fir filters. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages 545–548. IEEE, 1992.

[12] C. S. Burrus. Iterative reweighted least squares. *https://cnx.org/exports/92b90377-2b34-49e4-b26f-7fe572db78a1@12.pdf/iterative-reweighted-least-squares-12.pdf*.

[13] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[14] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 104–111. IEEE, 2009.

[15] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[16] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[17] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

[18] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, 1998.

[19] S. Cho and S. Lee. Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, volume 28, page 145. ACM, 2009.

[20] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 495–502. IEEE, 2011.

[21] T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman. Blur kernel estimation using the Radon transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 241–248. IEEE, 2011.

[22] W. Dong, G. Shi, and X. Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Transactions on Image Processing*, 22(2):700–711, 2013.

[23] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang. Compressive sensing via nonlocal low-rank regularization. *IEEE Transactions on Image Processing*, 23(8):3618–3632, 2014.

[24] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

[25] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[26] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[27] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[28] M. Elad, M. A. Figueiredo, and Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE Special Issue on Applications of Sparse Representation and Compressive Sensing*, 98(6):972–982, 2010.

[29] K. Engan, S. O. Aase, and J. H. Husøy. Multi-frame compression: Theory and design. *Signal Processing*, 80(10):2121–2140, 2000.

[30] M. Fadili and J.-L. Starck. Sparse representation-based image deconvolution by iterative thresholding. *Astronomical Data Analysis (ADA)*, 6:18, 2006.

[31] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.

[32] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 622–635. Springer, 2012.

[33] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I: fundamentals*, volume 305. Springer Science & Business Media, 2013.

[34] Z. Hu and M.-H. Yang. Good regions to deblur. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 59–72. Springer, 2012.

[35] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[36] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[37] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman. Image deblurring and denoising using color priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1550–1557. IEEE, 2009.

[38] S. Kim, Y.-W. Tai, S. J. Kim, M. S. Brown, and Y. Matsushita. Nonlinear camera response functions and image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25–32. IEEE, 2012.

[39] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.

[40] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 233–240. IEEE, 2011.

[41] D. Kundur and D. Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3):43–64, 1996.

[42] A. Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, 2006.

[43] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM Transactions on Graphics (TOG)*, volume 26, page 70. ACM, 2007.

[44] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647–1654, 2007.

[45] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1964–1971. IEEE, 2009.

[46] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2664. IEEE, 2011.

[47] L. Liu, B. Liu, H. Huang, and A. C. Bovik. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, 29(8):856–863, 2014.

[48] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proceedings of the IEEE Conference on Computer Vision (ICCV)*, pages 2272–2279. IEEE, 2009.

[49] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[50] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[51] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. Technical report, DTIC Document, 2007.

[52] J. Miskin and D. J. MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in independent component analysis*, pages 123–141. Springer, 2000.

[53] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[54] G. Monaci and P. Vandergheynst. Learning structured dictionaries for image representation. In *Proceedings of the IEEE International Conference on Image Processing*, volume 4, pages 2351–2354. IEEE, 2004.

[55] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[56] M. R. Osborne. *Finite algorithms in optimization and data analysis*. John Wiley & Sons, Inc., 1985.

[57] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[58] J. Pan, R. Liu, Z. Su, and X. Gu. Kernel estimation from salient structure for robust motion deblurring. *Signal Processing: Image Communication*, 28(9):1156–1170, 2013.

[59] J. Pan, R. Liu, Z. Su, and G. Liu. Motion blur kernel estimation via salient edges and low rank prior. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2014.

[60] J. Pan and Z. Su. Fast l_0-regularized kernel estimation for robust motion deblurring. *IEEE Transactions on Signal Processing Letters*, 20(9):841–844, 2013.

[61] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2909–2916. IEEE, 2014.

[62] S. Pertuz, D. Puig, and M. A. Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.

[63] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

[64] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 860–867. IEEE, 2005.

[65] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[66] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

[67] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[68] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 73. ACM, 2008.

[69] E. P. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *Proceedings of the Asilomar Conference on Signals, Systems & Computers*, volume 1, pages 673–678. IEEE, 1997.

[70] E. P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In *Bayesian inference in wavelet-based models*, pages 291–308. Springer, 1999.

[71] E. P. Simoncelli. 4.7 statistical modeling of photographic images. 2005.

[72] M. K. Steven. Fundamentals of statistical signal processing. *PTR Prentice-Hall, Englewood Cliffs, NJ*, 1993.

[73] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2013.

[74] Y.-W. Tai and S. Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24. IEEE, 2012.

[75] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *In IEEE Workshop on Statistical and Computational Theories of Vision*. Citeseer, 2003.

[76] J. Tropp, S. J. Wright, et al. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[77] S. Wang, L. Zhang, and Y. Liang. Nonlocal spectral prior model for low-level vision. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 231–244. Springer, 2013.

[78] Y. Wang and W. Yin. Compressed sensing via iterative support detection. *Rice University CAAM Technical Report TR09-30, http://www.caam.rice.edu/˜wy1/paperfiles/Rice_CAAM_TR09-30.pdf*, 2009.

[79] Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

[80] X. Wu, X. Zhang, and J. Wang. Model-guided adaptive recovery of compressive sensing. In *Proceedings of the Data Compression Conference (DCC)*, pages 123–132. IEEE, 2009.

[81] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 157–170. Springer, 2010.

[82] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l_0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, page 174. ACM, 2011.

[83] L. Xu, S. Zheng, and J. Jia. Unnatural $l_0$sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1107–1114. IEEE, 2013.

[84] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[85] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *ACM Transactions on Graphics (TOG)*, volume 27, page 74. ACM, 2008.

[86] L. Zhang, A. Deshpande, and X. Chen. Denoising vs. deblurring: HDR imaging techniques using moving cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 522–529. IEEE, 2010.

[87] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.

[88] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 612–619. IEEE, 2013.

요약

블라인드 영상 선명화는 흐린 영상으로부터 고품질의 영상을 복원하는 것을 목적으로 한다. 영상 선명화는 문제의 공식화, 제한법의 적용, 최적화 방법과 많은 관련을 갖기 때문에 최근에 많은 관심을 얻고 있다. 최적화 관점에서 블라인드 영상 선명화는 아주 심한 역문제로 수많은 해를 가질 수 있다. 그 중에서 고품질의 영상을 해로 얻기 위해서는 효과적인 제한법을 적용해야 한다. 이 논문에서 우리는 블라인드 영상 선명화를 위해서 비근접 제한법을 제안한다. 첫째, 우리는 커널을 가중치로 적용한 유사도로 얻은 비근접 부분영상의 사용을 제안한다. 커널에 의한 비근접 부분영상을 이용해서, 비슷한 비근접 부분영상은 합성곱을 적용하더라도 여전히 비슷할 것이라는 사전지식을 제한으로 반영한다. 이런 제한법은 블러-커널 추정을 향상시킨다. 둘째, 우리는 비근접 저차수 영상의 사용을 제안한다. 비근접 저차수 영상은 비근접 부분영상을 엮어서 얻는다. 이 비근접 저차수 영상을 이용하여, 우리는 선명한 이미지가 비근접 저차수 영상과 비슷할 것이라는 사전지식을 제한으로 가한다. 비근접 저차수 영상은 그 속성상 잡음을 적게 가지고 있기 때문에 이런 제한은 잡음이 적은 선명한 영상을 추정하도록 한다. 우리는 제안한 방법을 정량적, 정성적으로 평가하여 전통적인 블라인드 영상 선명화 방법과 비교하였다. 정량적 평가를 위해서는 오차의 제곱합, 최대 신호대 잡음비, 구조유사도를 측정하였다. 잡음이 포함되지 않은 흐린 영상에서는 우리의 방법이 다른 방법에 비해 대체로 우수함을 보여주었다. 특히 우리의 방법은 구조물에서 선명함을 보여주었고, 평평한 영역에서 부드러운 결과를 보여주었다. 잡음이 포함된 흐린 영상에서는 우리의 방법이 다른 방법에 비해 크게 우수함을 보여주었다. 대부분의 방법이 블커-커널 추정에 실패하는데 반해, 우리의 방법은 잡음을 극복하고 성공적으로 블러-커널을 추정하였고, 잡음이 적은 고품질의 영상을 복원하였다.

주요어: 영상선명화, 비근접 제한법, 비근접 저차수 영상, 블러-커널 추정
학번: 2008-22937